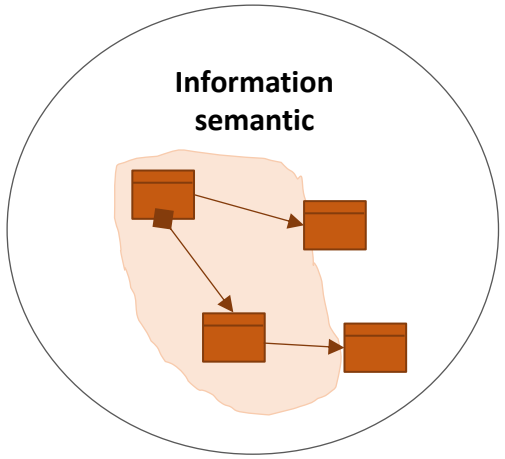
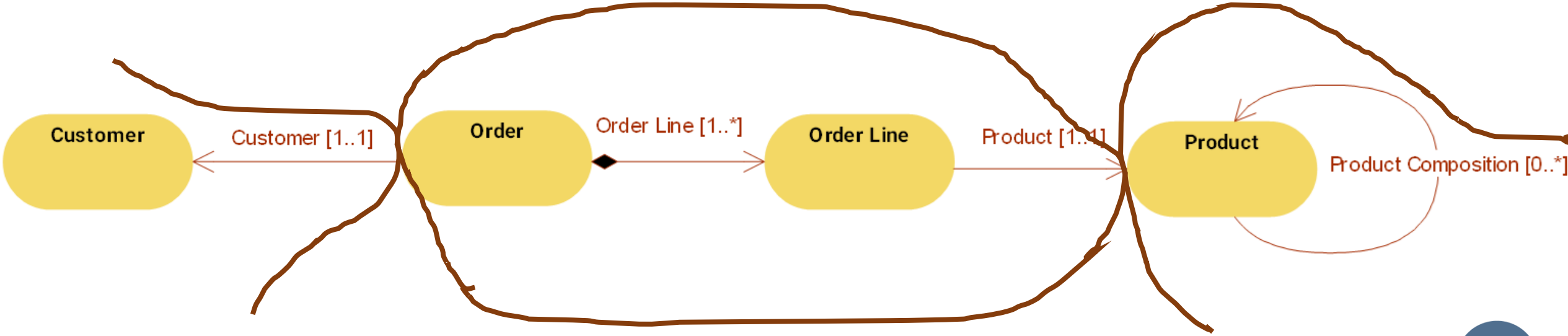
Systemic Framework for Enterprise Architecture & Transformation

Ontology Modeling

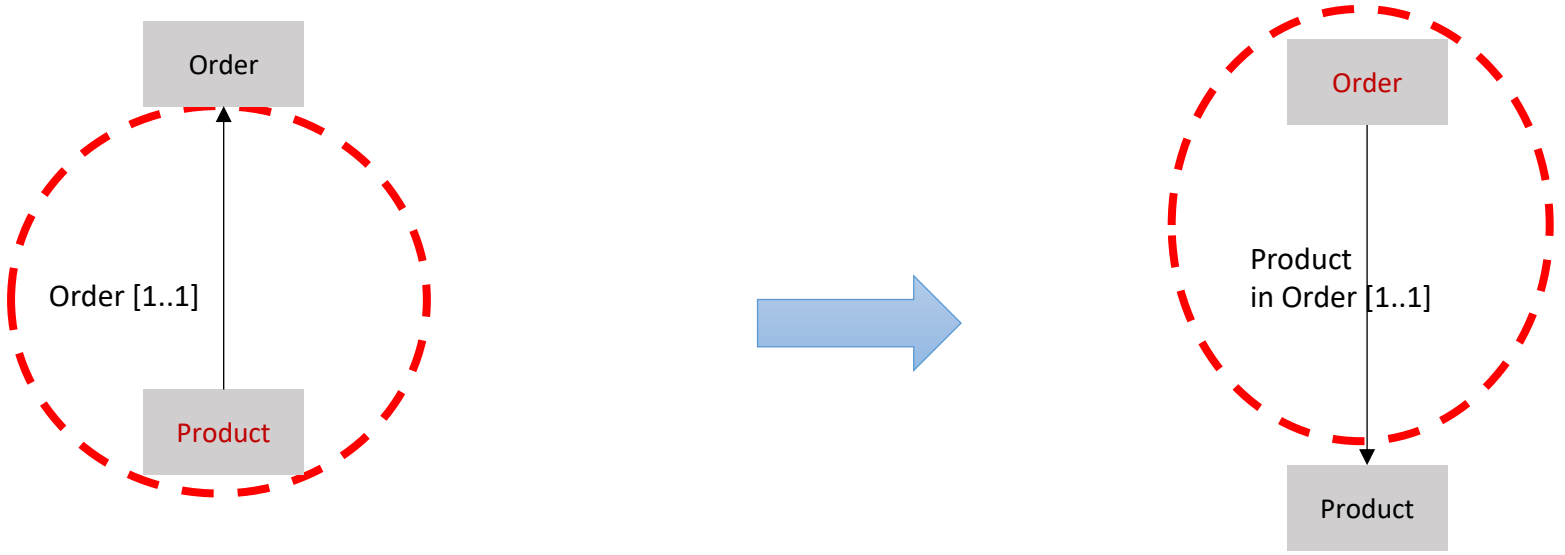
Information semantic : Business Object Scopes



- The trick of digitalization consists in introducing gaps into continuums, thereby creating boundaries.
  - Jesper Hoffmeyer
- Scope Identification is the mantra of architecture: what is in, what is out ?
- Directed relationships help in drawing boundaries.



Whole/part relationships: Relationship direction rule

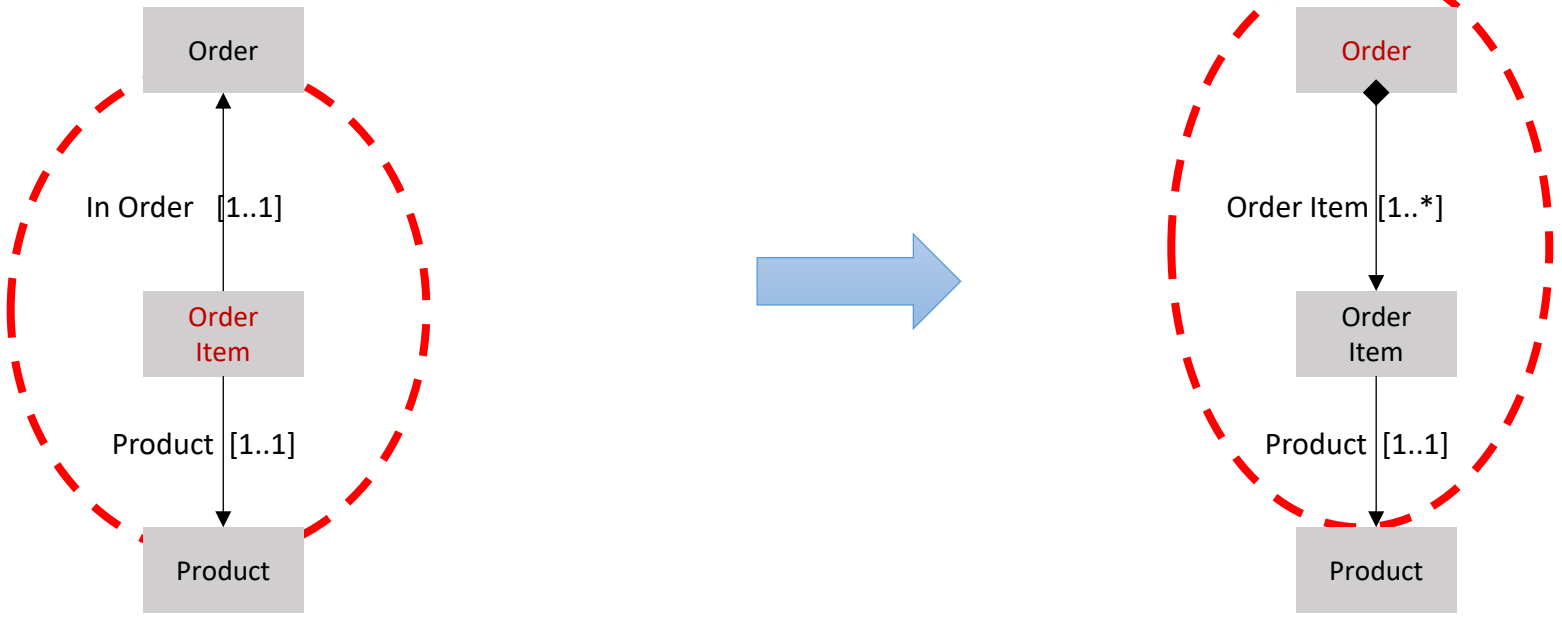


Products are not made of Orders…

… but Orders are always about Products.

Tip : Finding the right direction might seem tough at the beginning. A good way to know is to wonder *“Who would loose information if I cut the link ?”,* or *“Who knows the other ?”*

Whole/part relationships: Relationship direction rule



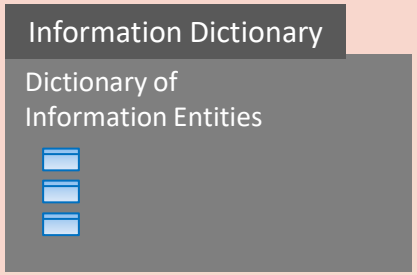
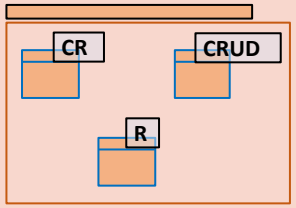
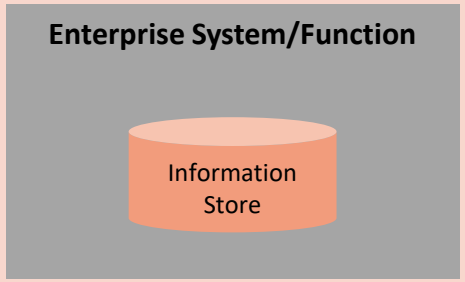
Order Items are not made of Orders…

… but Orders are always about Order Items.

Tip : Order Items are specific kind of entities that cannot exist without their whole. It doesn't mean they are defines by their whole. The whole/part relationship between whole and such entities is a composition relationship (black diamond).

The rule for finding the right direction steel apply: *"Who would loose information if I cut the link ?".* Without Order Items, Orders are undefined.

Definitions : Information Dictionary, Entities, Domains, Stores

Information Dictionary	Information Domain	Information Store
		
Dictionary of Information Entities defining an information landscape. Information Dictionaries are often confused with Information Domains. Information Dictionaries are mere body of knowledge. They have a container role that assert the existence of information entities that they include.	Group of Information Entities that are required to be accessed or updated for a business domain to operate. Information Domains define how Information Entitles can be accessed in their context: [CRUD]. For instance, the “Sales Information Domain” is composed of the Customer entity with a [CRUD] access, the Order entity with a [CRUD] access, and the Product entity with a [R] access. Information Domains are usually under the responsibility of one ore more Data Architect and one Data Custodian.	An information store references an entity domain necessary for its operation. Information stores are of two kinds: • Local Information Store that represents a store under the control of an enterprise system. • External Information Store that represents a subset of an Information Store manipulated by an application or application system.

PS : Information Views and advanced Structured Entities are only available in MEGA Information Architecture

Information Allocation Consistency

- For a given Information Dictionary, the *Information Allocation Report* lists all its Information Entities, their managing systems along with CRUD characteristics.
- The split of information entities in Information Domains and the decomposition of enterprise systems in sub-systems shall be done accordingly so that Information Domain boundaries and enterprise system boundaries coincide as much as possible.

Information Entity	Create	Read	Update	Delete
Product	• Marketing System • Purchasing System	• Marketing System	• Marketing System • Purchasing System	• Marketing System
Purchase Order	• Purchasing System	• Purchasing System	• Purchasing System	• Purchasing System

Information Usage report

- For a given Information Dictionary, the *Information Usage Report* lists all its Information Entities, their managing systems and their CRUD agents or sub-agents.
- A derived report delivers the same content layout but its entry point is an agent internal structure.
- Information Entities are all information entities of information stores on the root agent internal structure and of its components.
- Agents in columns are the sub-agents of the root agent.

Information dictionary					
Information Entity	**Managing System**	**Create**	**Read**	**Update**	**Delete**
Information Entity 1	System S1	App 1 App X	App 2 App Y	App 3	App 7
Information Entity 2	System SA System SB				

Enterprise System Information Domains and Entities					
Information Entity	**Managing System**	**Create**	**Read**	**Update**	**Delete**
Information Entity 1	App System S1	App 1 App X	App 2 App Y	App 3	App 7
Information Entity 2	App System SA App System SB				

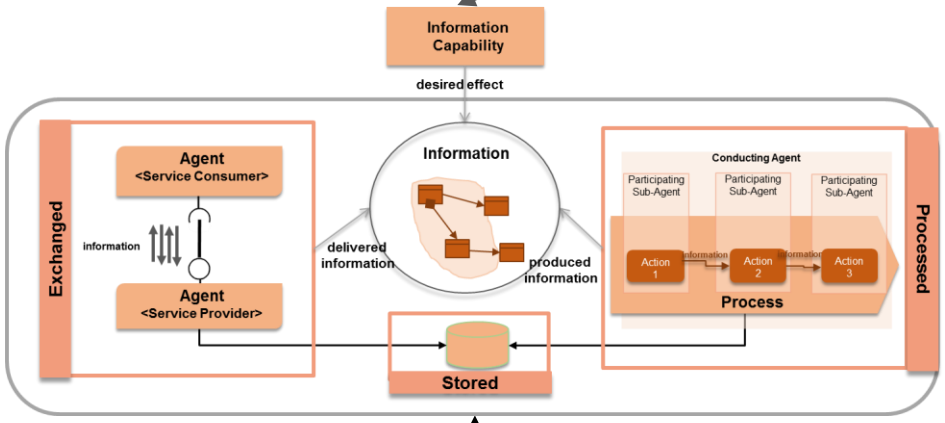
Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

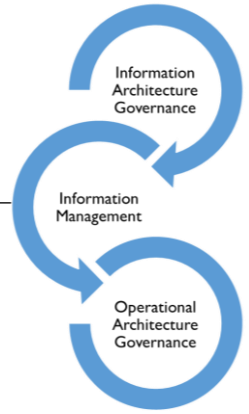
Information Architecture - Stakeholder Perspectives

Information Architecture

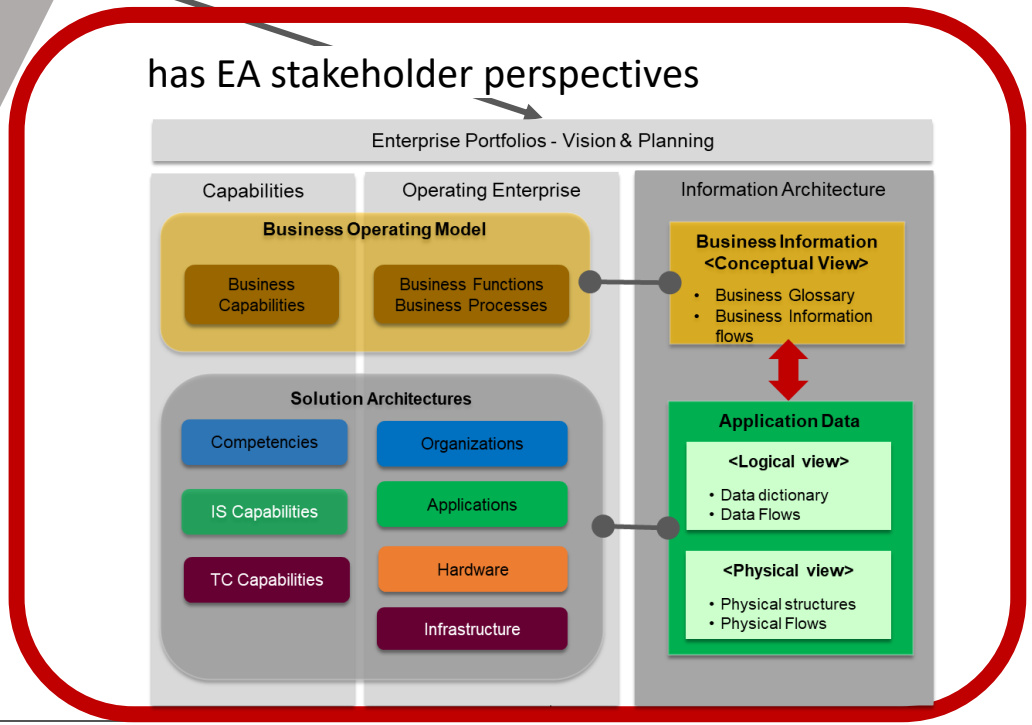
has architecture aspects



is under agile management



has EA stakeholder perspectives

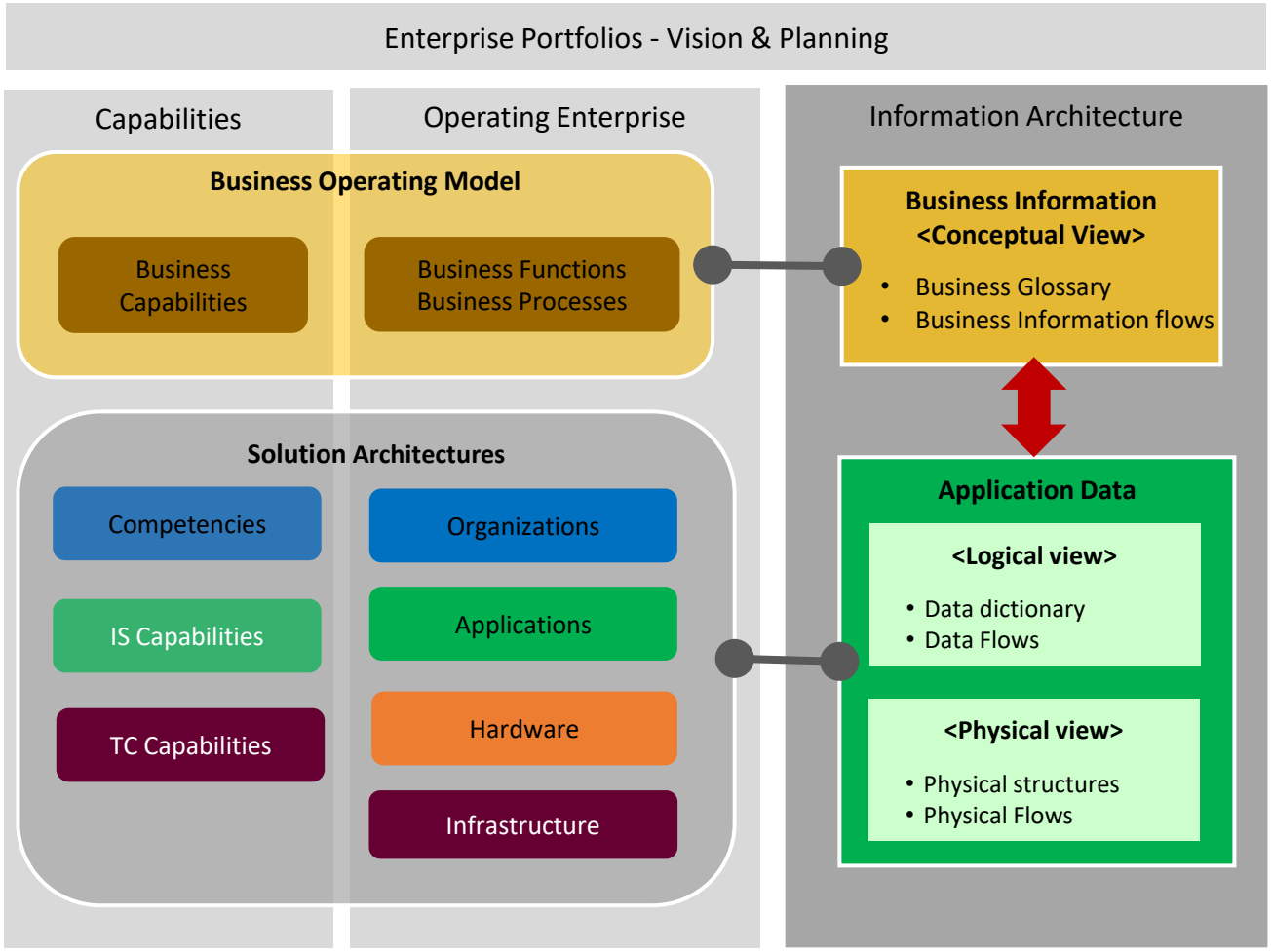


Information Architecture - Stakeholder Perspectives

- Information can be studied with different EA perspectives:

- The business level:
  - Information descriptions are used to depict the business domain, independently of the way information is physically managed: through paper work or through information systems.
  - This level is also sometimes called the "**conceptual view**" where the glossary is defined.

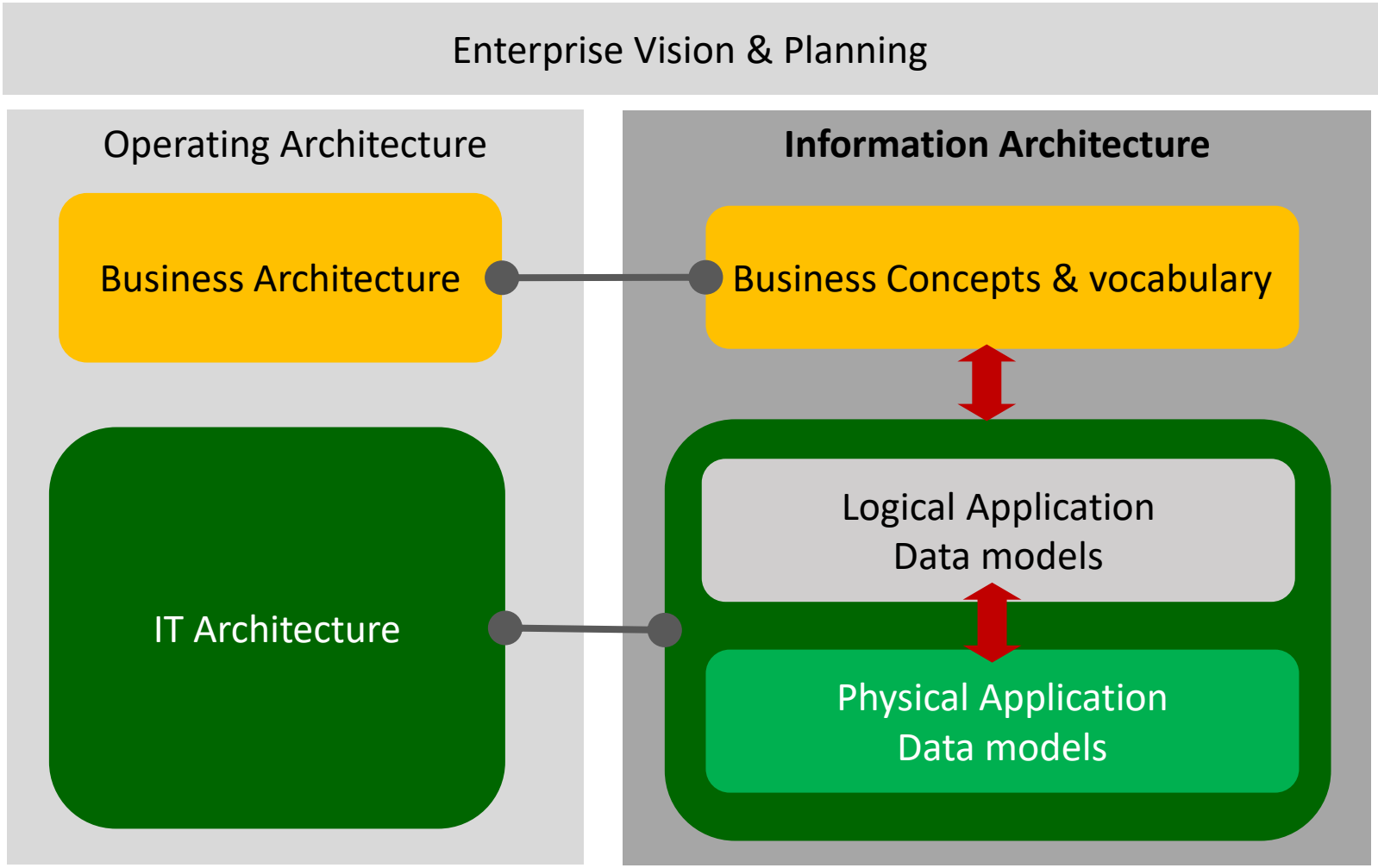
- The application level:
  - Information descriptions are used to depict the subset of the business information domain that is stored and manipulated by IT systems to be consumed the organization.
  - Application data are considered according to two viewpoints:
  - **A logical viewpoint**: a view of data independent of implementation.
  - **A physical viewpoint**: a view of data tied to a specific data format (relational, XML, file records, etc.).



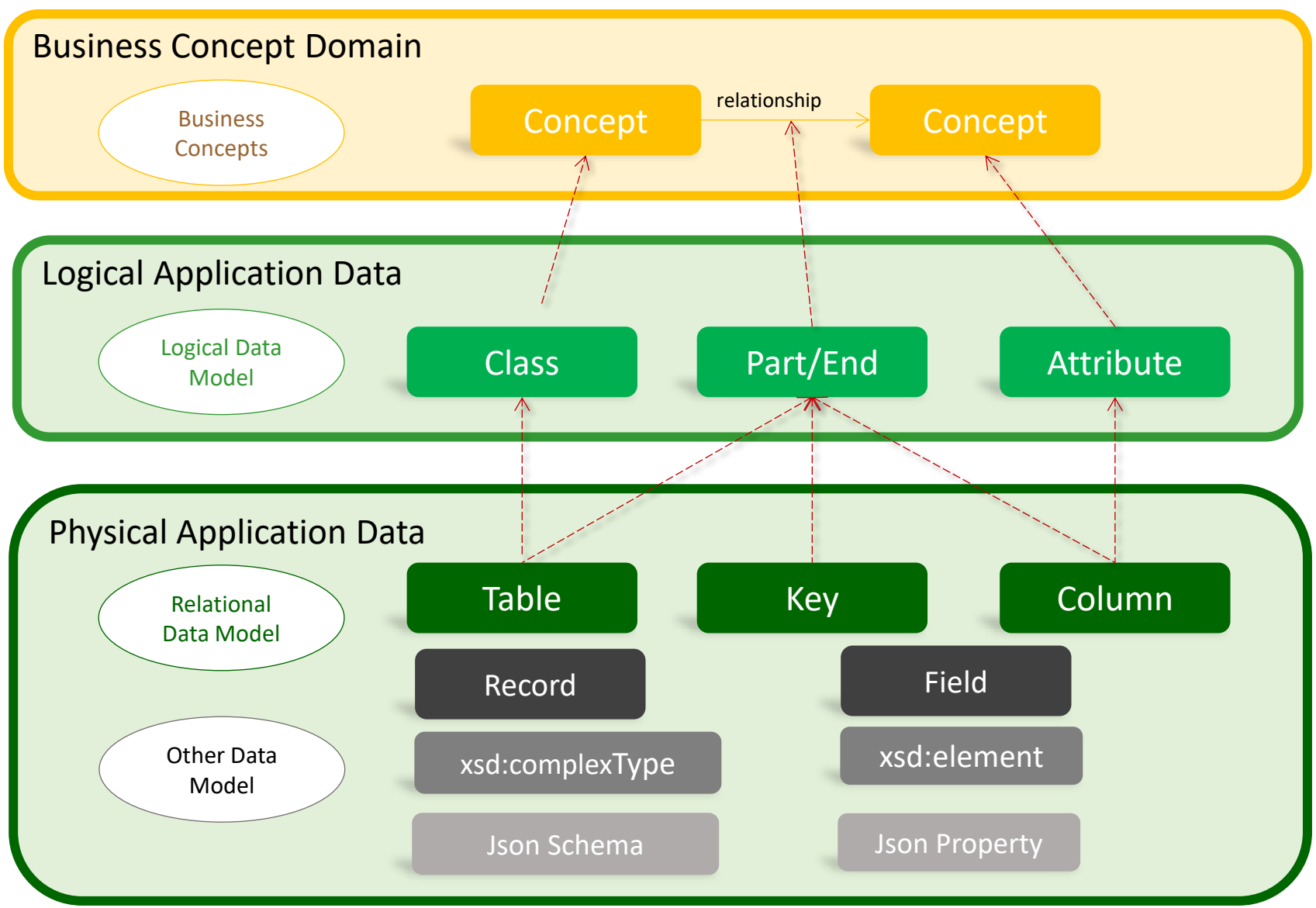
Information Architecture - Stakeholder Perspectives (simplified)

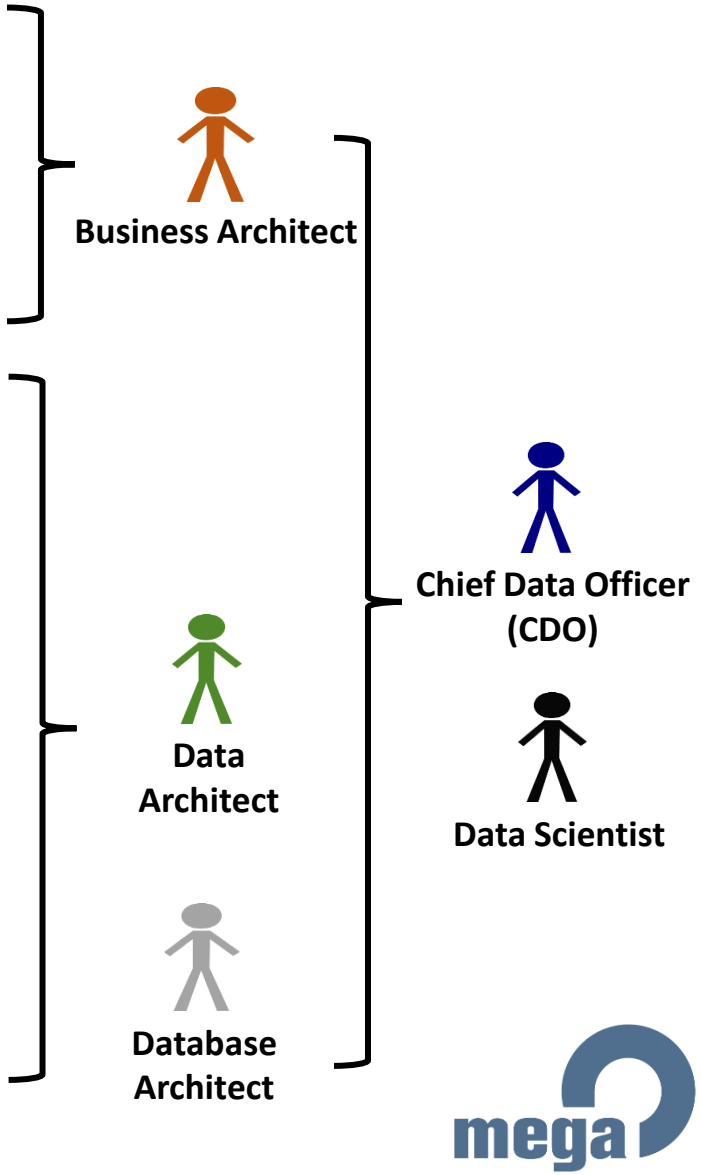
- The business level:
  - Information descriptions are used to depict the business domain, independently of the way information is physically managed: through paper work or through information systems.

- The application level:
  - Information descriptions are used to depict the subset of the business information domain that is stored and manipulated by IT systems to be consumed the organization.
  - Application data are considered according to two viewpoints:
  - **A logical viewpoint**: a view of data independent of implementation.
  - **A physical viewpoint**: a view of data tied to a specific data format (relational, XML, file records, etc.).



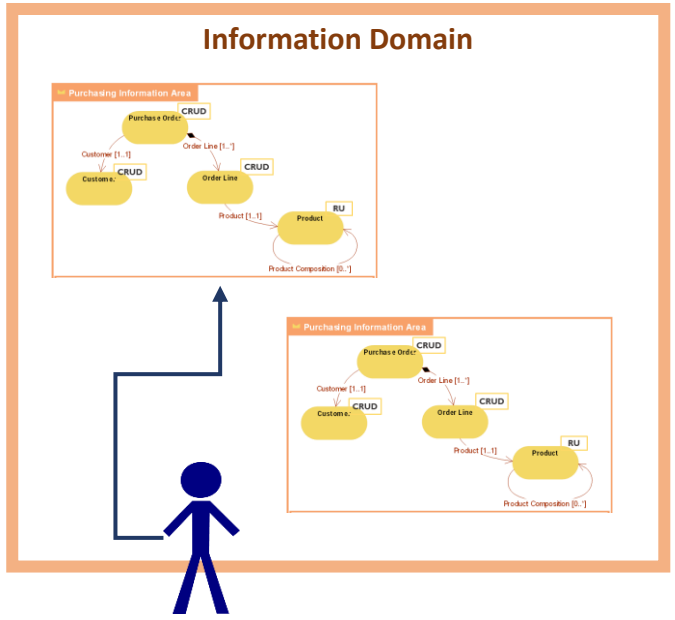
Conceptualization levels & Stakeholders





City Planning of Information

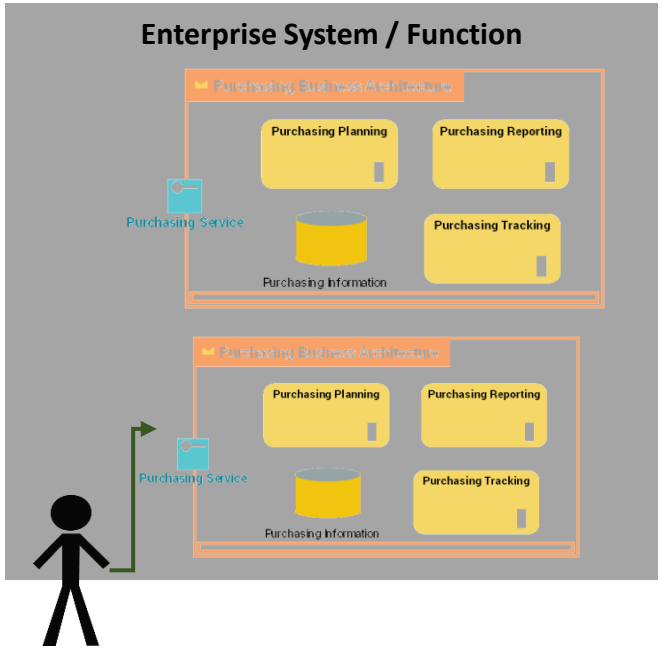
- The split of information entities in Information Domains and the decomposition of enterprise systems in sub-systems shall be done accordingly so that Information Domain boundaries and enterprise system boundaries coincide as much as possible.
- The goal is to obtain modular information systems that enable the agile evolution of enterprise information solutions.
- The governance of proper scoping of Information Domains shall ensure that:
  - Information Domains and Systems decomposition are kept as consistent as possible.
  - The various EA stakeholders - Information & System architect & Designers - are kept in a governance loop to ensure a global quality of information.



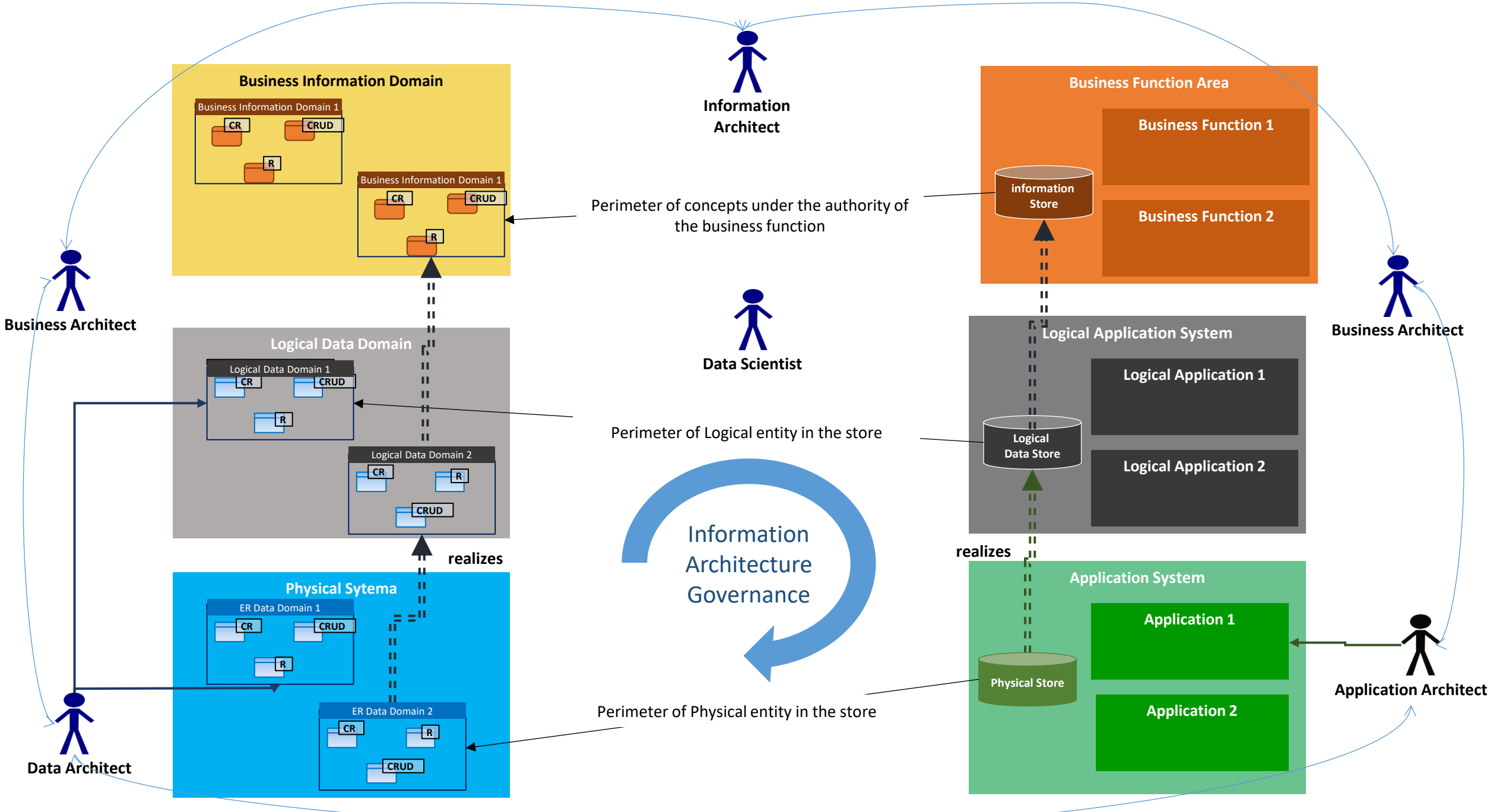
- **Information Architect [Designer]**
  Data Steward [Owner]

Architecture Consistency

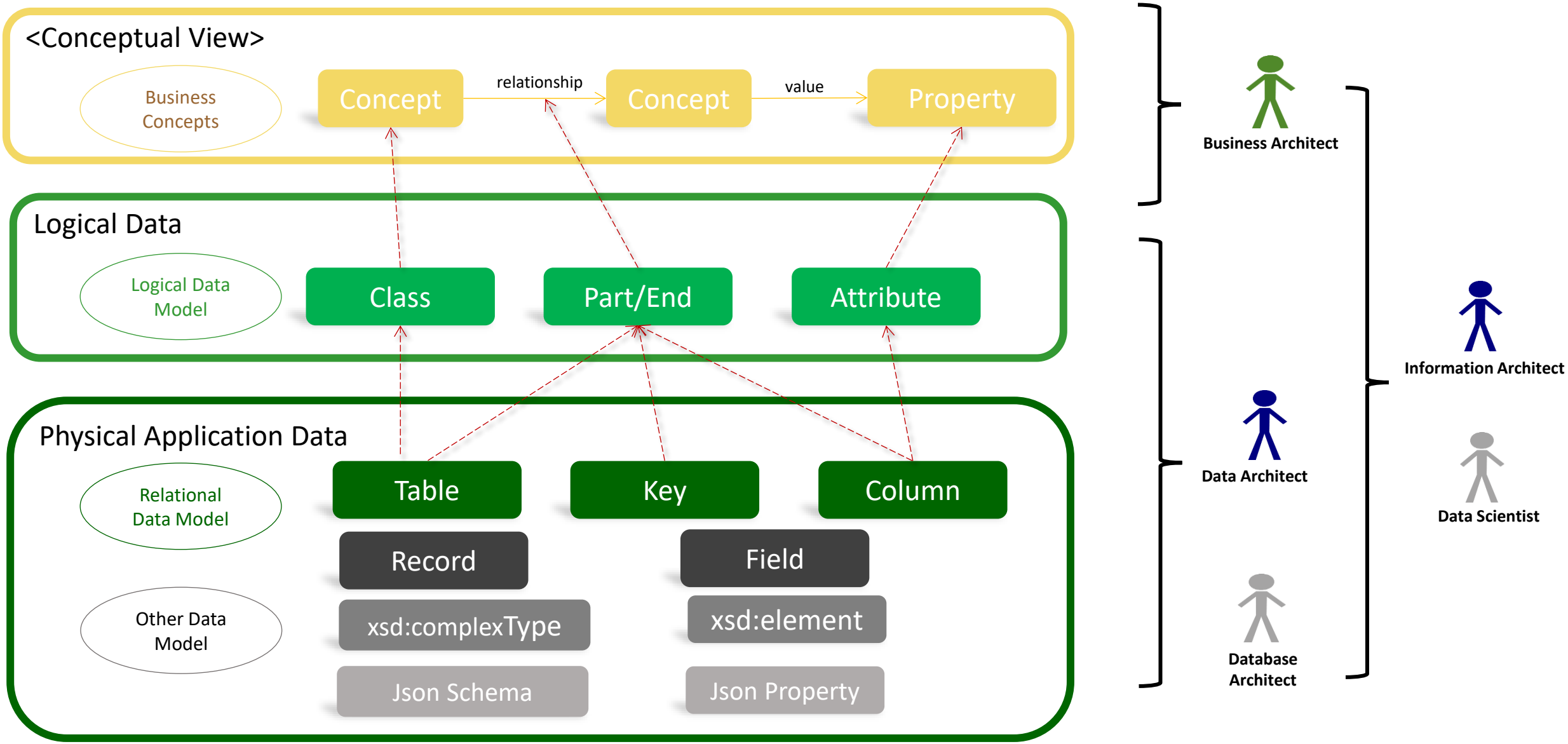
Governance Consistency



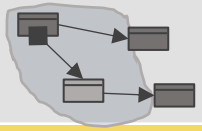
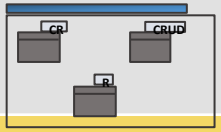
- **System Architect [Designer]**
- **System Asset Manager [Owner]**


Information Architect
Business Information Domain
Business Information Domain 1
CR
CRUD
R
Business Information Domain 1
CR
CRUD
R
Business Function Area
Business Function 1
information Store
Business Function 2
Perimeter of concepts under the authority of the business function
Business Architect
Business Architect
Data Scientist
Logical Data Domain
Logical Data Domain 1
CR
CRUD
R
Logical Data Domain 2
CR
R
CRUD
Logical Application System
Logical Application 1
Logical Data Store
Logical Application 2
Perimeter of Logical entity in the store
realizes
realizes
Information Architecture Governance
Physical Sytema
ER Data Domain 1
CR
CRUD
R
ER Data Domain 2
CR
R
CRUD
Application System
Application 1
Physical Store
Application 2
Perimeter of Physical entity in the store
Application Architect
Data Architect

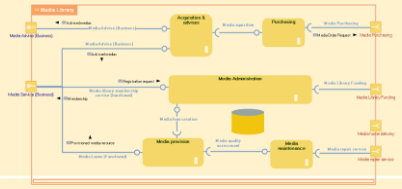
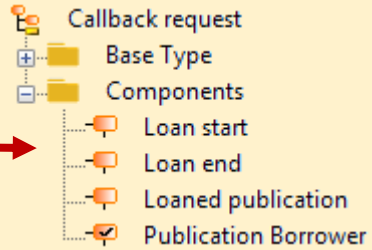
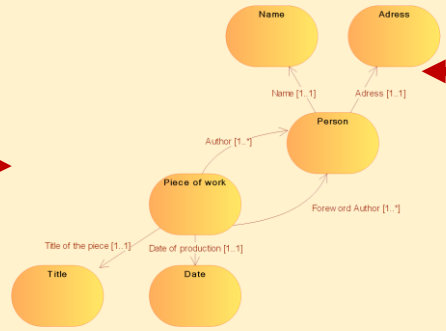
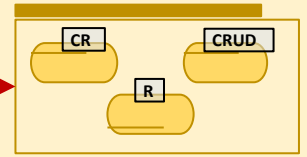
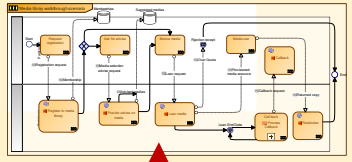
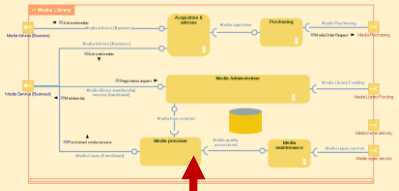
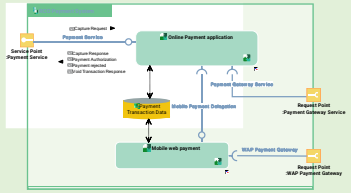
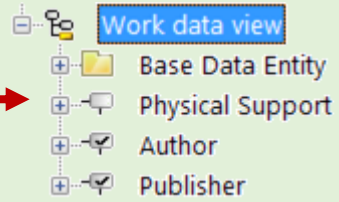
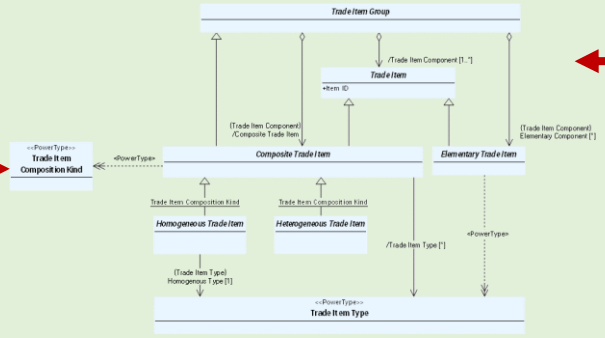
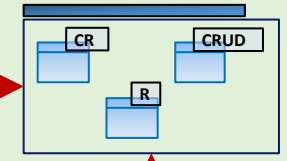
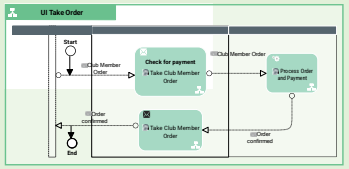
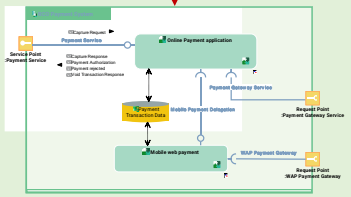
Conceptualization levels Concepts



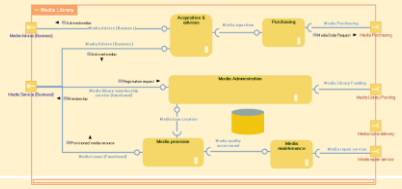
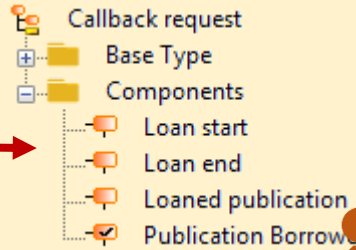
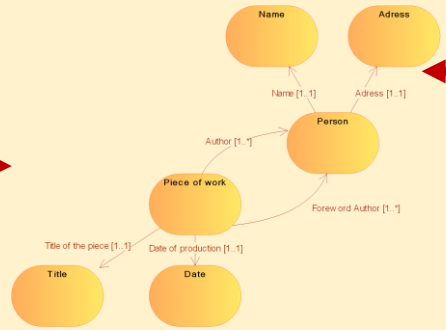
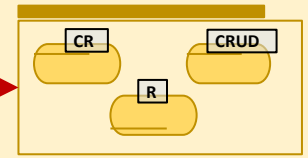
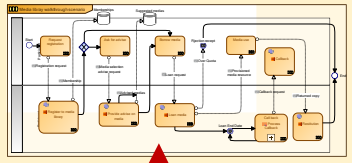
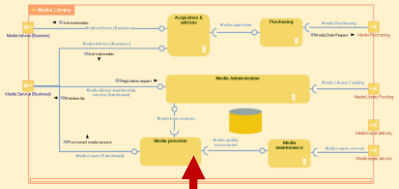
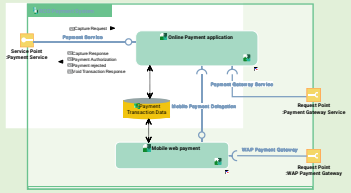
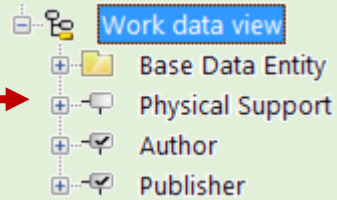
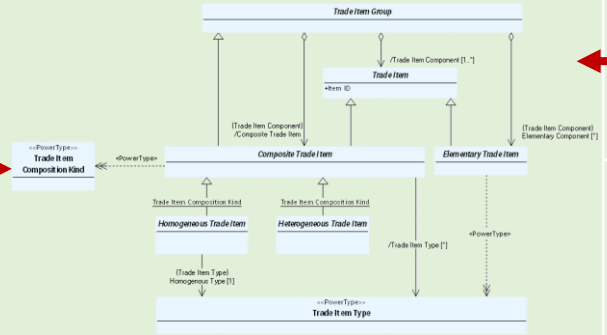
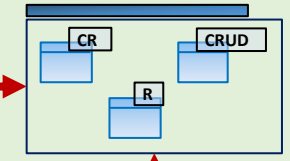
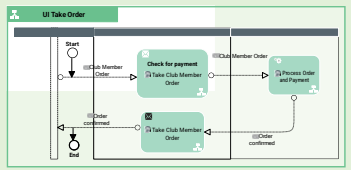
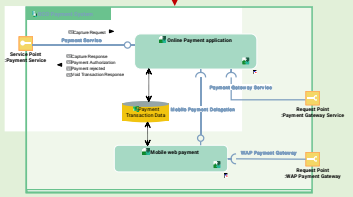
Information Pattern across EA layers

	Information Dictionary	Information Entity	Information Domain	Information Store
				
Concepts	Business Dictionary (Dictionary of concepts)	Concept	Concept Domain	Business Information Store
Logical Data	Package (Dictionary of Classes)	Class	Logical Data Domain	Logical Data Store
Relational Data	Database (Dictionary of Tables)	Table	Relational Schema	Physical Data Store
NoSQL Data	Meta Data Package (Dictionary of NoSQL entities)	Meta Dataset	No SQL Schema	

Data Architecture Aspects across Layers

	Information Flows	Information Views	Information Entities	Information Responsibility Areas
Business Layer	**Business Information Flows**	**Business Information Views**	**Business Information Reference Model**	**Business Information Domain**
	Interactions 		Glossary & Terms 	 Grouping
	Process flows 			 Storage
Application Layer	**Application Data Flows**	**Application Data Views**	**Application Data Entities**	**Application Data Domain**
	Interactions 		Class & relationship 	 Grouping
	Process Flows 			 Storage

Data Architecture Aspects across Layers - Stakeholders

	Information Flows	Information Views	Information Reference Model	Information Responsibility Areas
Business Architect Business Layer	**Business Information Flows**	**Business Information Views**	**Business Information Reference Model**	**Business Information Domain**
	Interactions 		Glossary & Terms 	 Grouping
Business Architect	Process flows 			 Storage
	Application Data Flows	**Application Data Views**	**Application Data Entities**	**Application Data Domain**
Application Layer Data Architect	Interactions 		Data dictionary 	 Grouping
	Process Flows 			 Storage

Information Architect

Application Architect

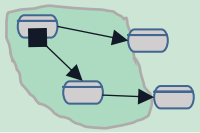
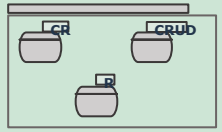
Data Scientist

Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

Conceptual Mondel

- Conceptual Modeling has two purposes:
  - Describe what the business concepts that make up business glossaries.
  - Describe business concept domains boundaries and dependencies between business concepts domains.
    - Managed concepts.
    - Exchanged Information.

	Information Dictionary	Information Entity	Information Domain	Information Store
				
Business Information	Subject Area (Dictionary of concepts)	Concept	Business Concepts Domain	Business Information Store

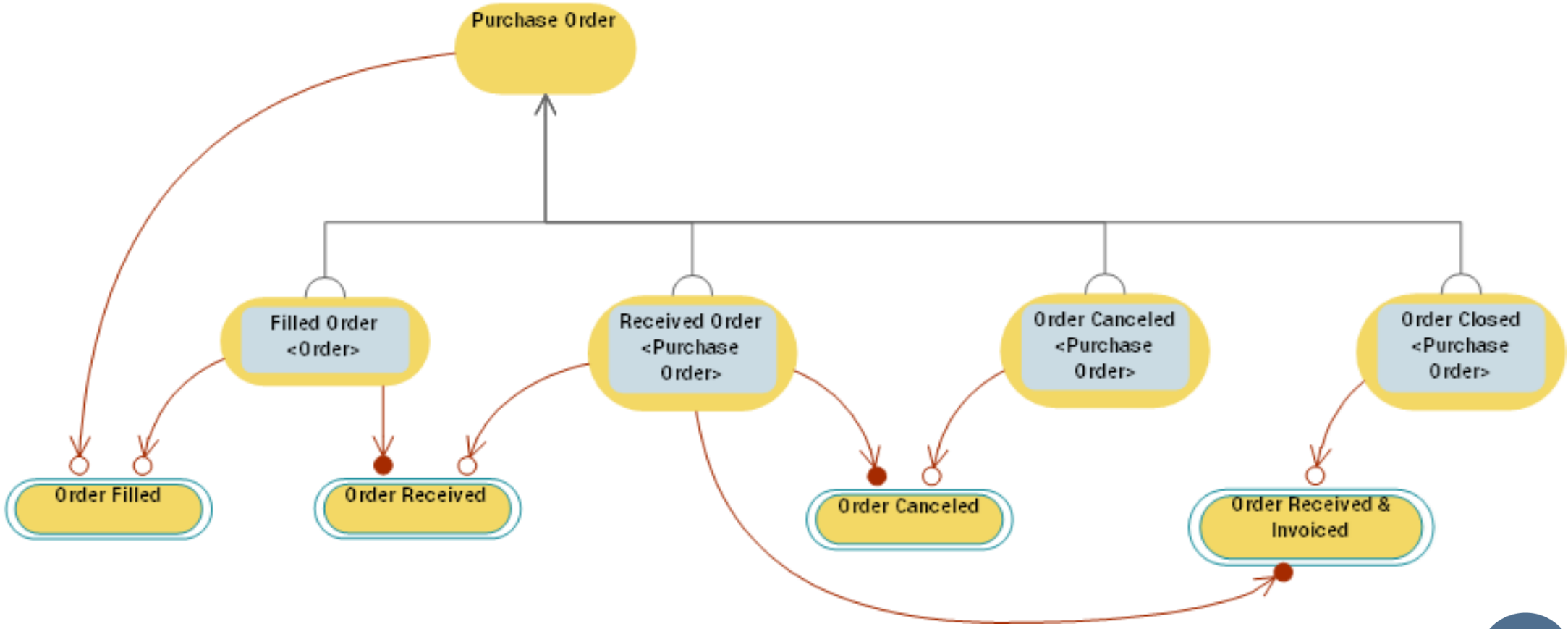
Information SEMANTIC: the world is not static

… introducing time in Business Information Models

Concept

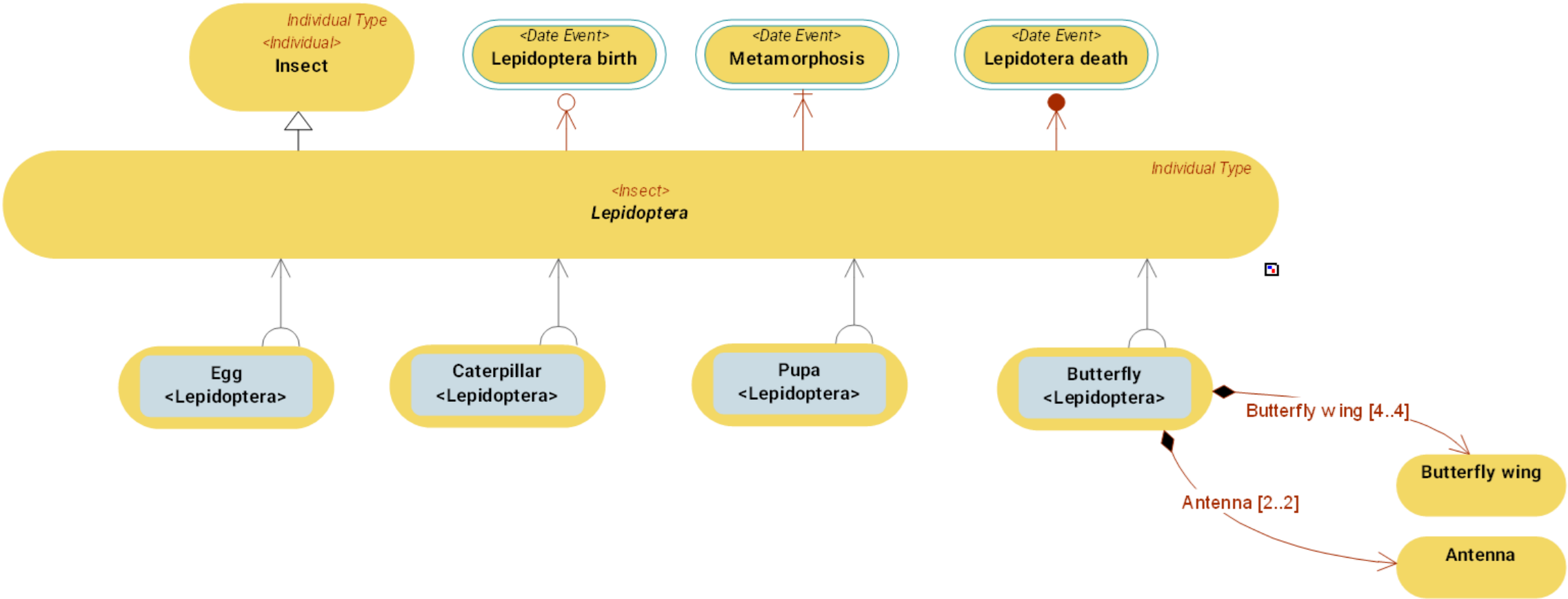
State

Event



States are entities

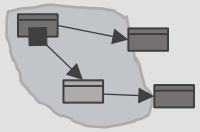
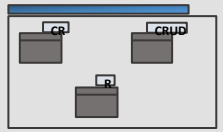
- Only butterflies (state of the Lepidoptera) have wings
- => States can have relationships, as any entities.
- => Some relationships are only valid for specific states (butterfly wings below).



Agenda

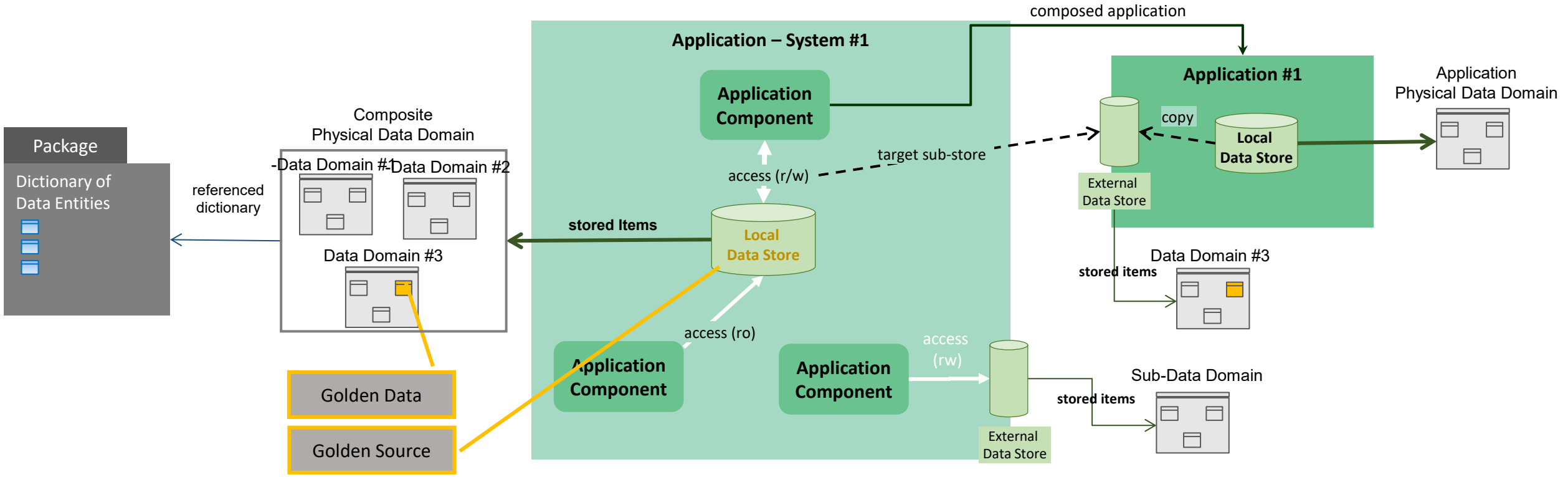
- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

Application Information Architecture

	Data Dictionary	Data Entity	Data Domain	Information Store
				
Logical Data	Package (Dictionary of Classes)	Class Data View	Logical Data Domain	Logical Data Store
Relational Data	Database (Dictionary of Tables)	Table Physical View	Relational Schema	Physical Store
NoSQL Data	Meta Data Package	Meta DataSet Meta DataView	NoSQL Schema	

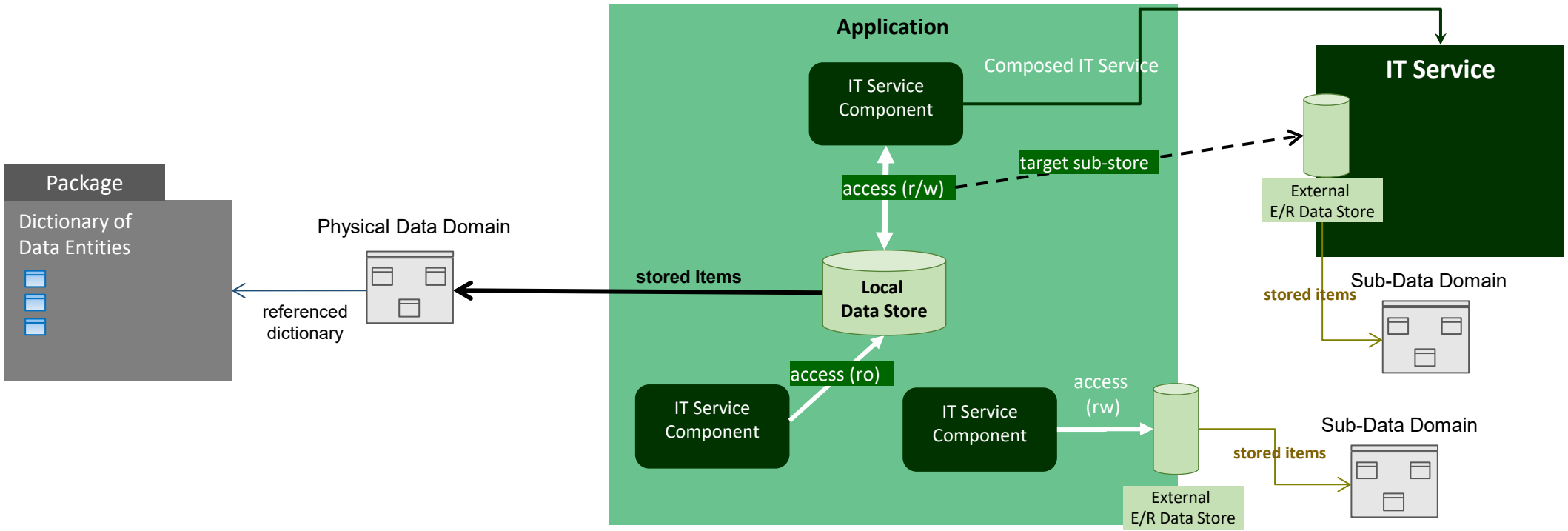
Application System, Applications & Data Stores

- An application system provides or manages Data Stores:
  - Local Data Stores are local (owned and managed) by the considered application or application system.
  - External Data Stores are sub-data stores needed by other applications but not managed by these applications.
- Application Components declare their need for shared data through External Data Stores:
- Application Components are granted access to shared data stores through data access relationships read/write mode or a read only mode.



Application, IT Services & Data Stores

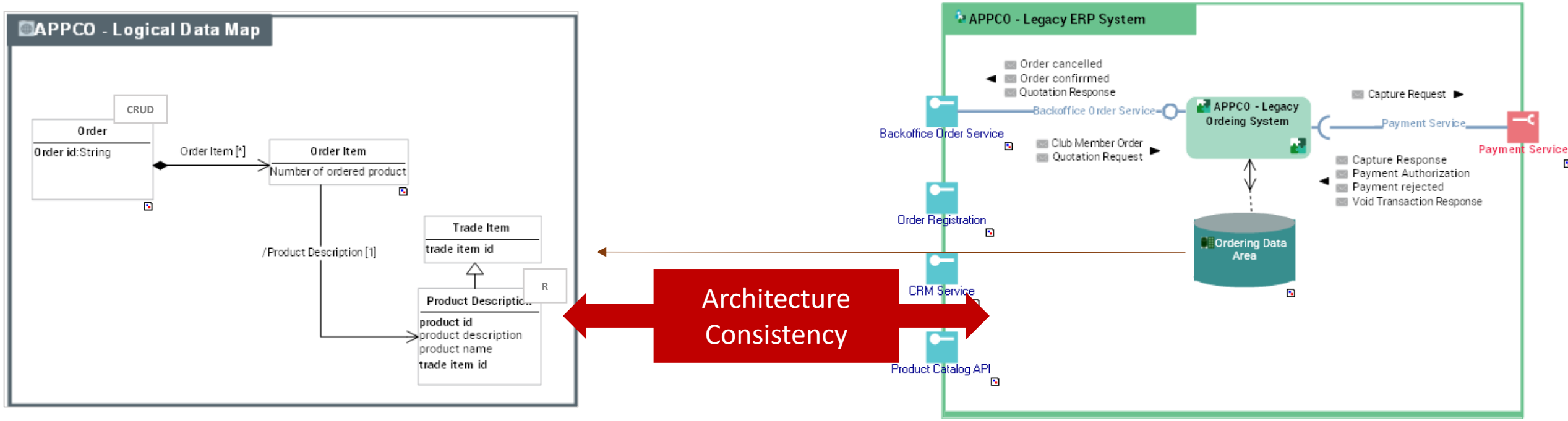
- An application references or manages Data Stores:
  - Local Data Stores are local (owned and managed) by the considered application or application system.
  - External Data Stores are sub-data stores needed by other applications but not managed by these applications.
- IT Service Components declare their need for data through External Data Stores:
- IT Service Components are granted access to data stores through data access relationships read/write mode or a read only mode.
- Data stores can be either E/R Data stores and Relational Data Stores.



Data Allocation

- The split of data entities in Data Domains and the decomposition of enterprise systems in sub-systems shall be done accordingly so that Information Domain boundaries and enterprise system boundaries coincide as much as possible.
- The goal is to obtain modular information systems that enable the agile evolution of enterprise information solutions.

- The allocation of Information Domains to Enterprise Systems is done through “Information Stores”.
  - An Information Store expresses the fact that its associated “Information Domain” is under the control of the considered Enterprise System.
    - For instance the “Purchasing Information Domain” references all business concepts required for the “Purchasing Function” to be able to operate.



Information Usage report

- For a given data catalog, the data usage report lists all its data entities, their managing system and their CRUD applications.
- A derived report delivers the same content layout but its entry point is a software system.
  - Data Entities are all data entities of data stores on the root software systems and of its components.
  - Applications in columns are the sub-components of the root system.

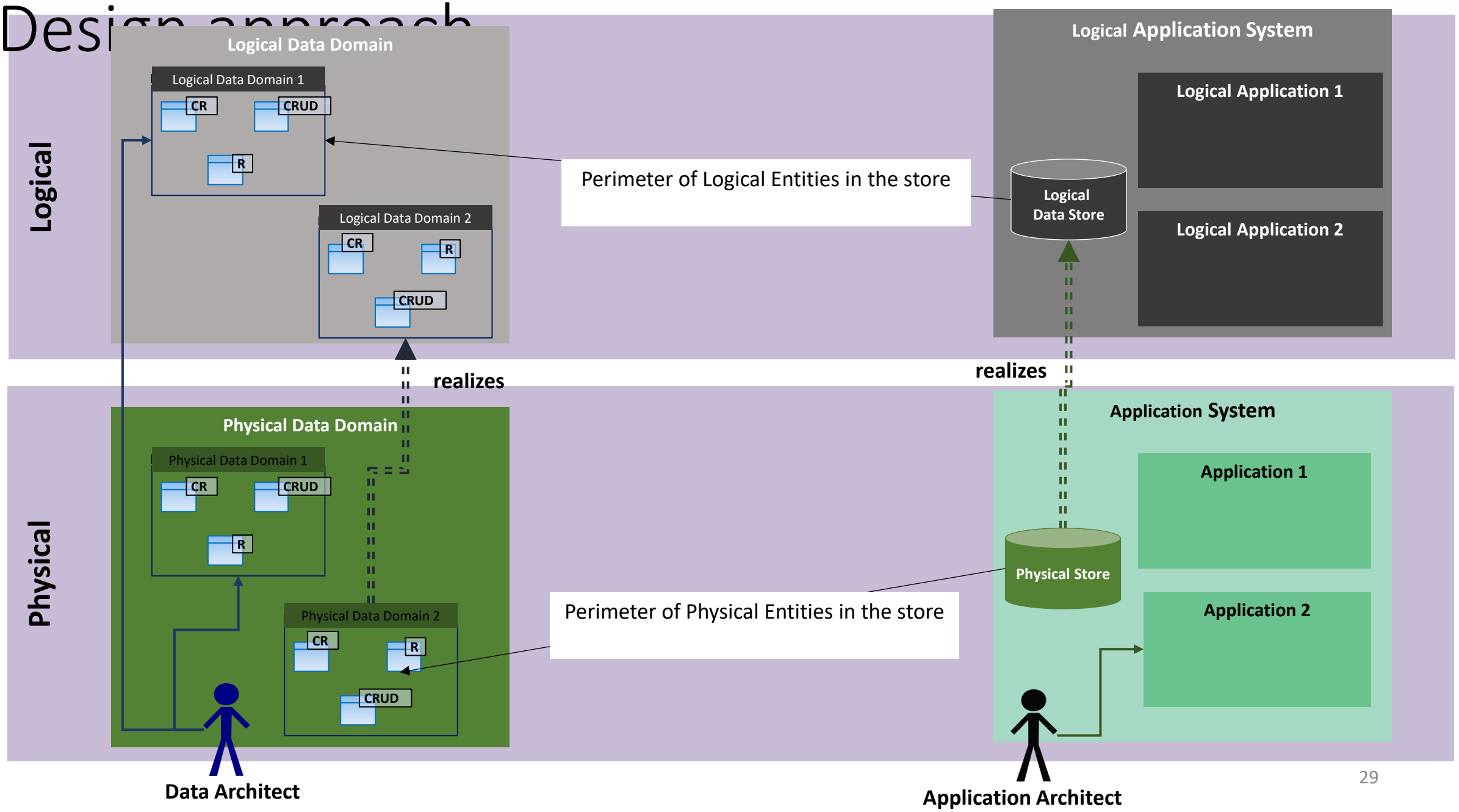
Data catalog: Data Dictionary X (Data Dictionary = Database in current MEGA)

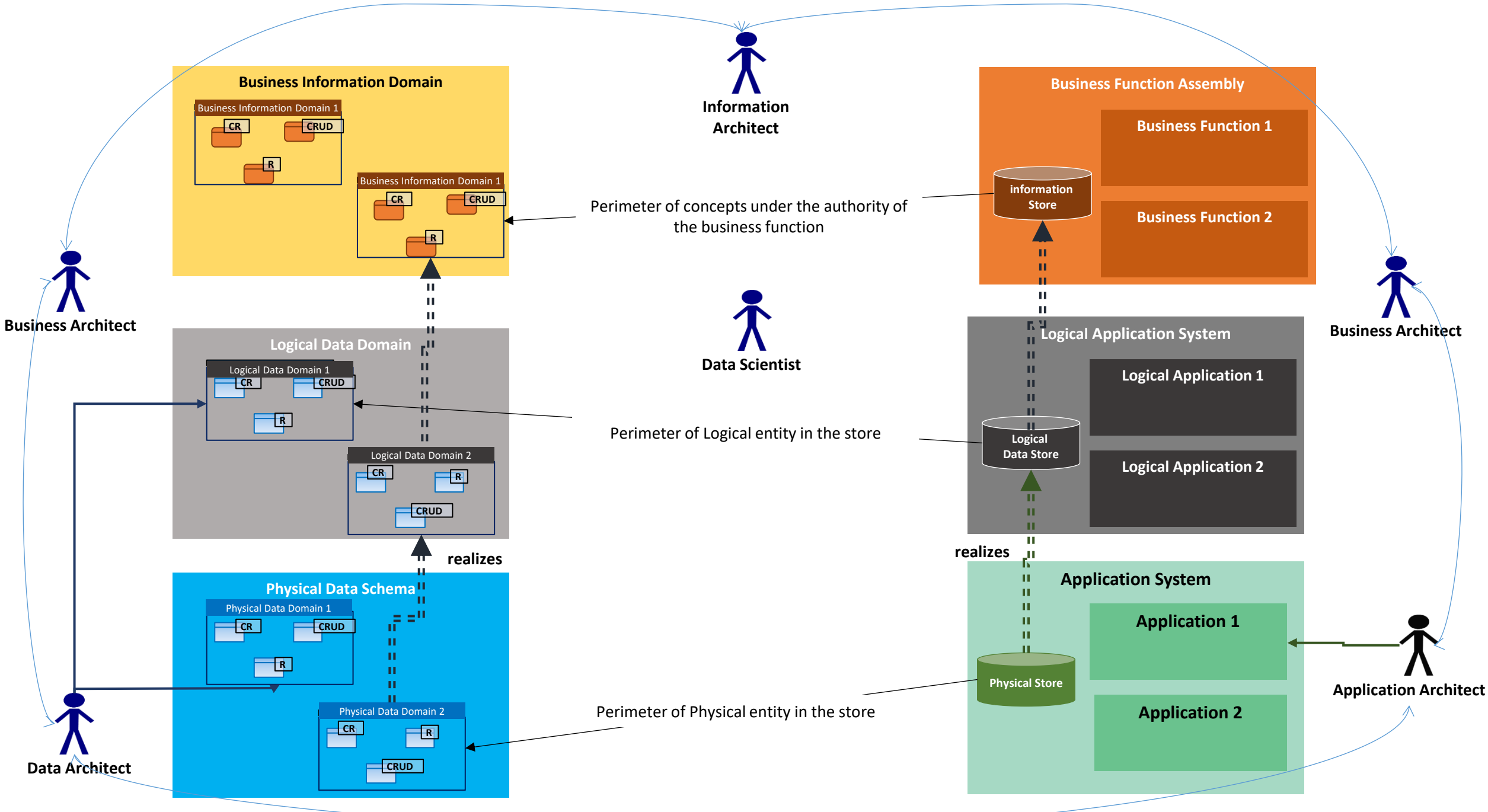
Data Entity	Managing Application	Create	Read	Update	Delete
Data Entity 1	App System S1	App 1 App X	App 2 App Y	App 3	App 7
Data Entity 2	App System SA App System SB				

System Data : Software System in MEGA (Abstraction of Logical Architecture & Application System)

Data Entity	Managing Application	Create	Read	Update	Delete
Data Entity 1	App System S1	App 1 App X	App 2 App Y	App 3	App 7
Data Entity 2	App System SA App System SB				

Design approach




Information Architect
Business Information Domain
Business Information Domain 1
CR
CRUD
R
Business Information Domain 1
CR
CRUD
R
Business Function Assembly
Business Function 1
information Store
Business Function 2
Business Architect
Perimeter of concepts under the authority of the business function
Business Architect
Data Scientist
Logical Data Domain
Logical Data Domain 1
CR
CRUD
R
Logical Data Domain 2
CR
R
CRUD
Perimeter of Logical entity in the store
Logical Application System
Logical Application 1
Logical Data Store
Logical Application 2
realizes
realizes
Physical Data Schema
Physical Data Domain 1
CR
CRUD
R
Physical Data Domain 2
CR
R
CRUD
Application System
Application 1
Physical Store
Application 2
Application Architect
Data Architect
Perimeter of Physical entity in the store

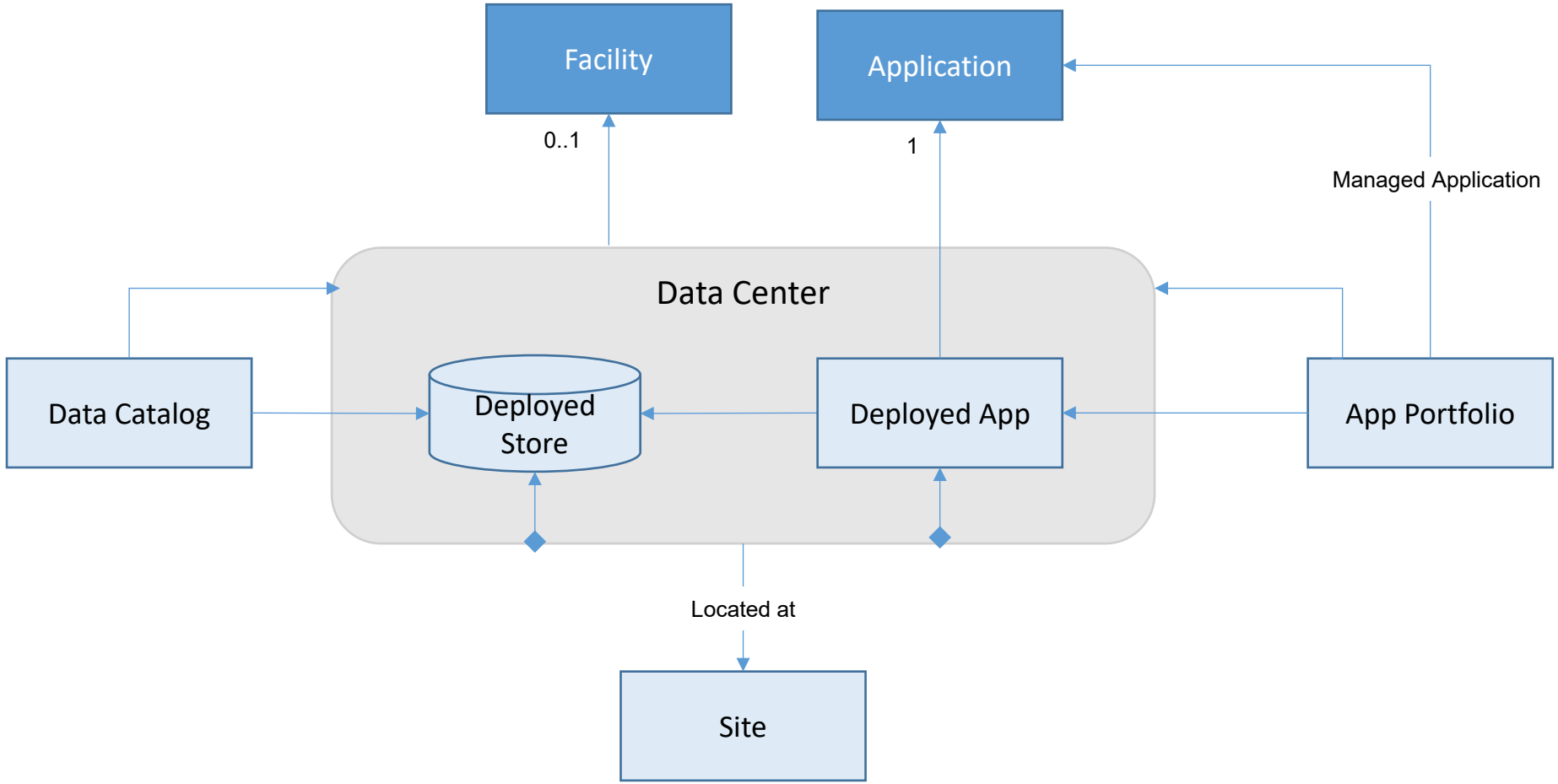
Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

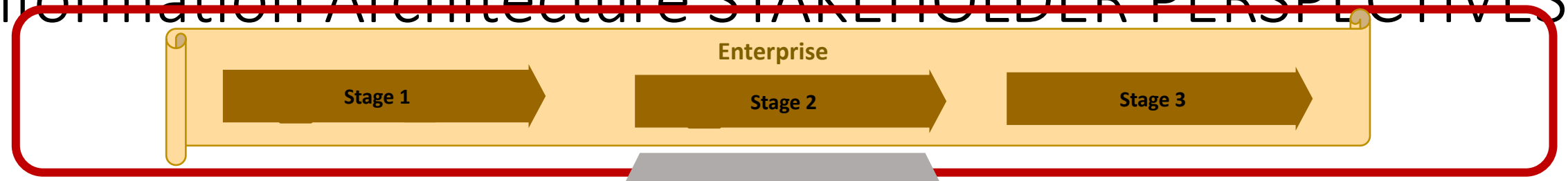
Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

Deployed Store

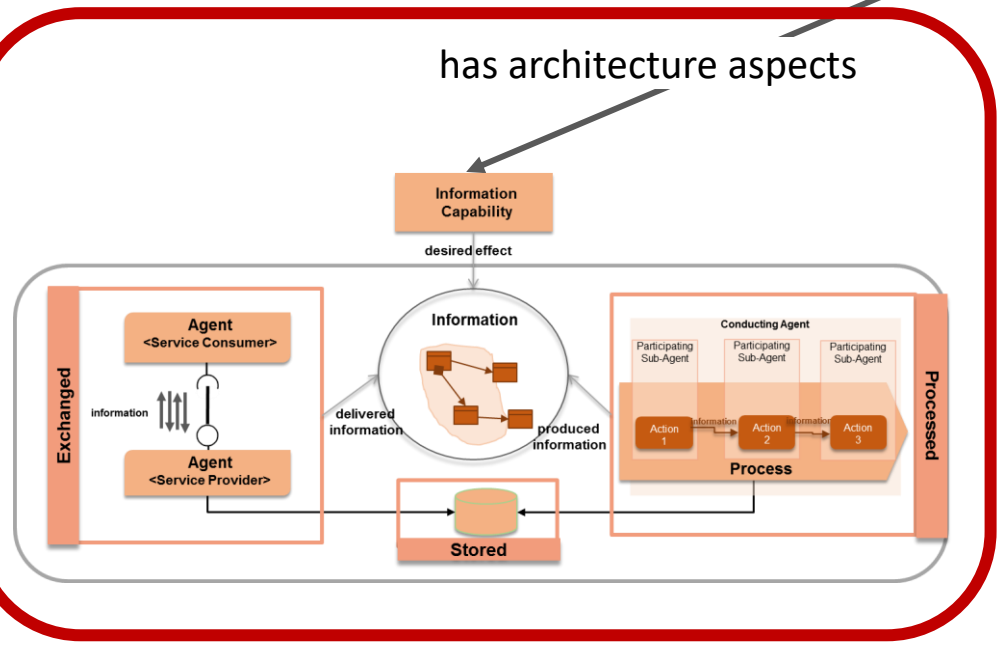


Information Architecture STAKEHOLDER PERSPECTIVES

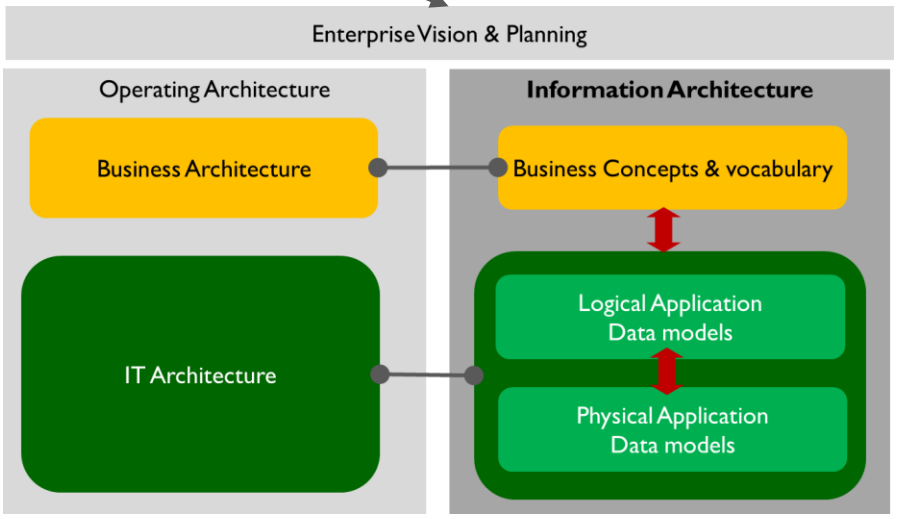


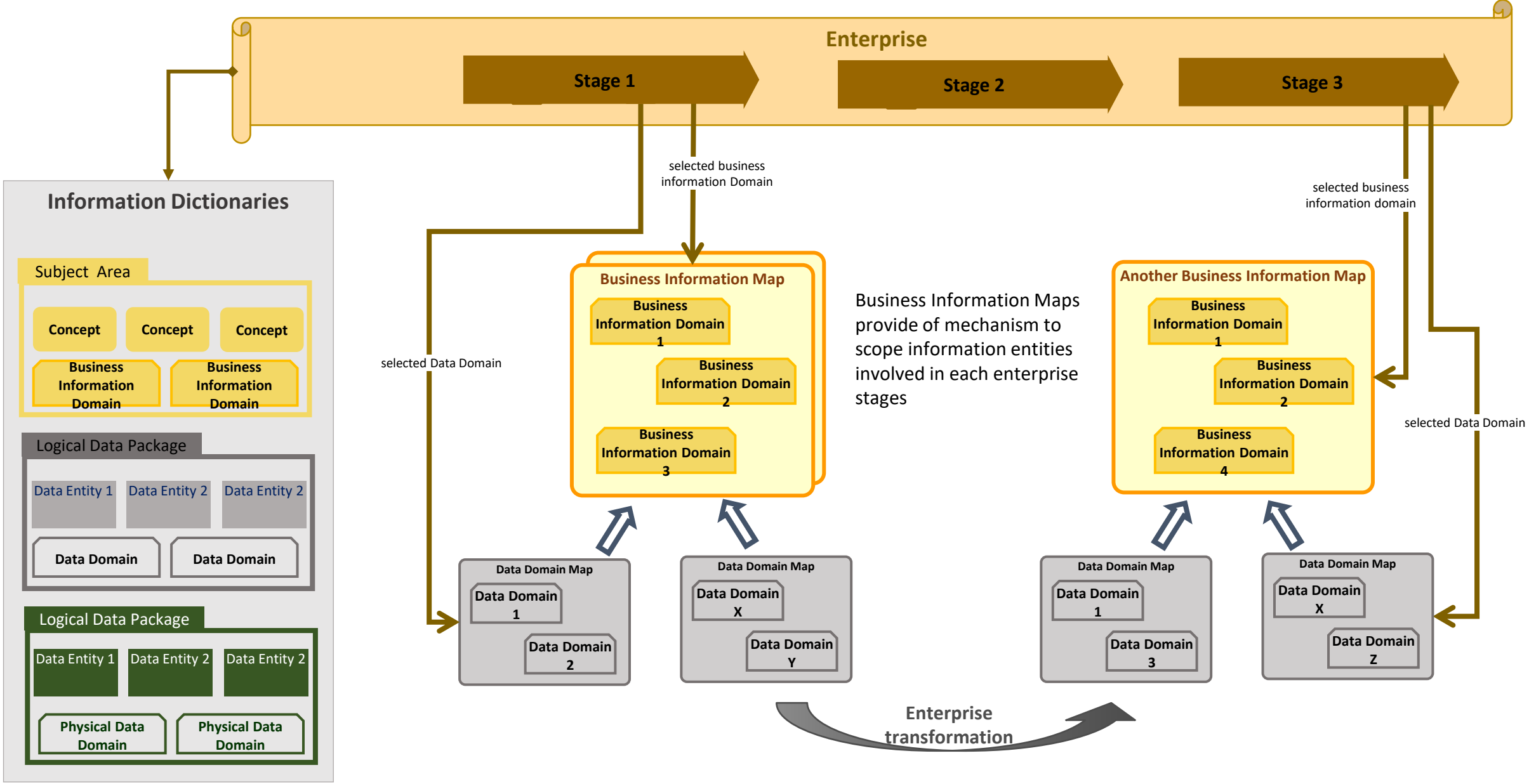
Information Architecture

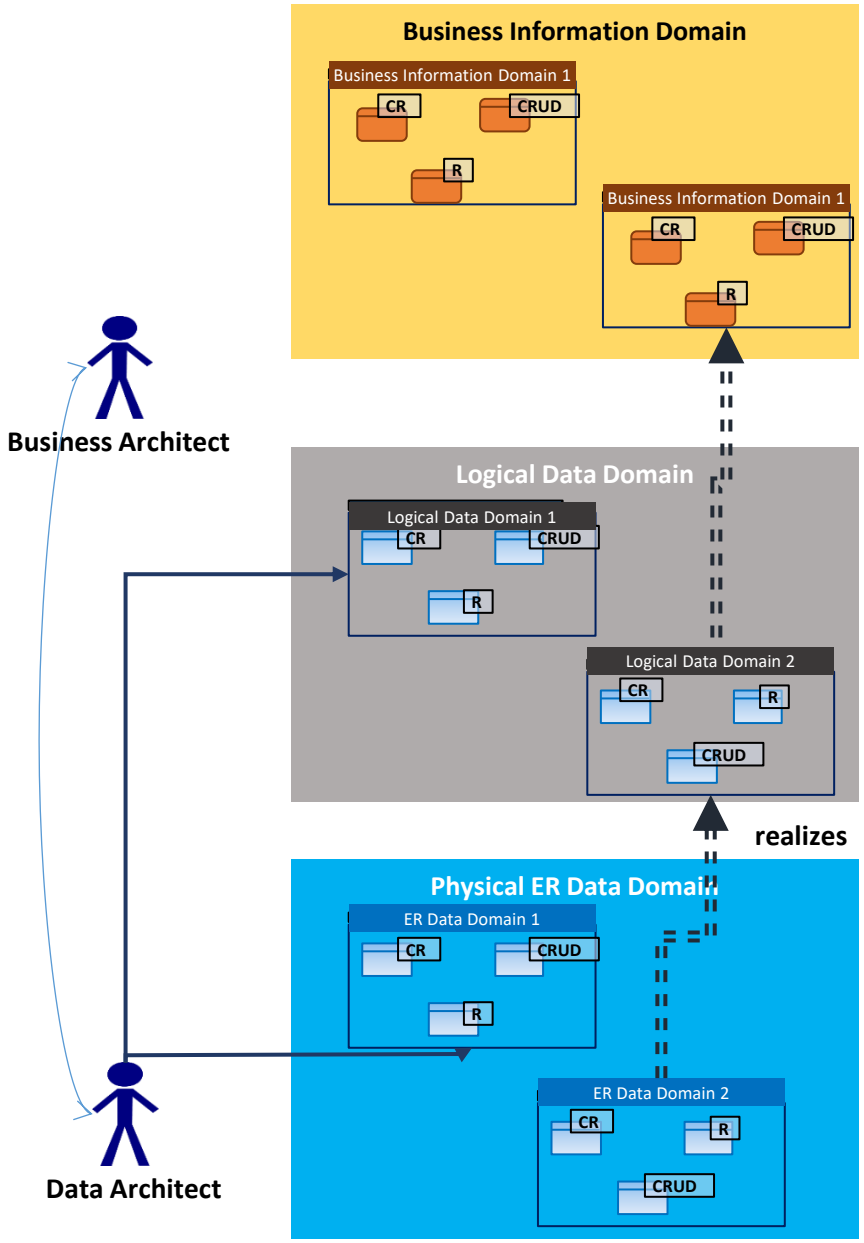
has architecture aspects



has EA stakeholder perspectives



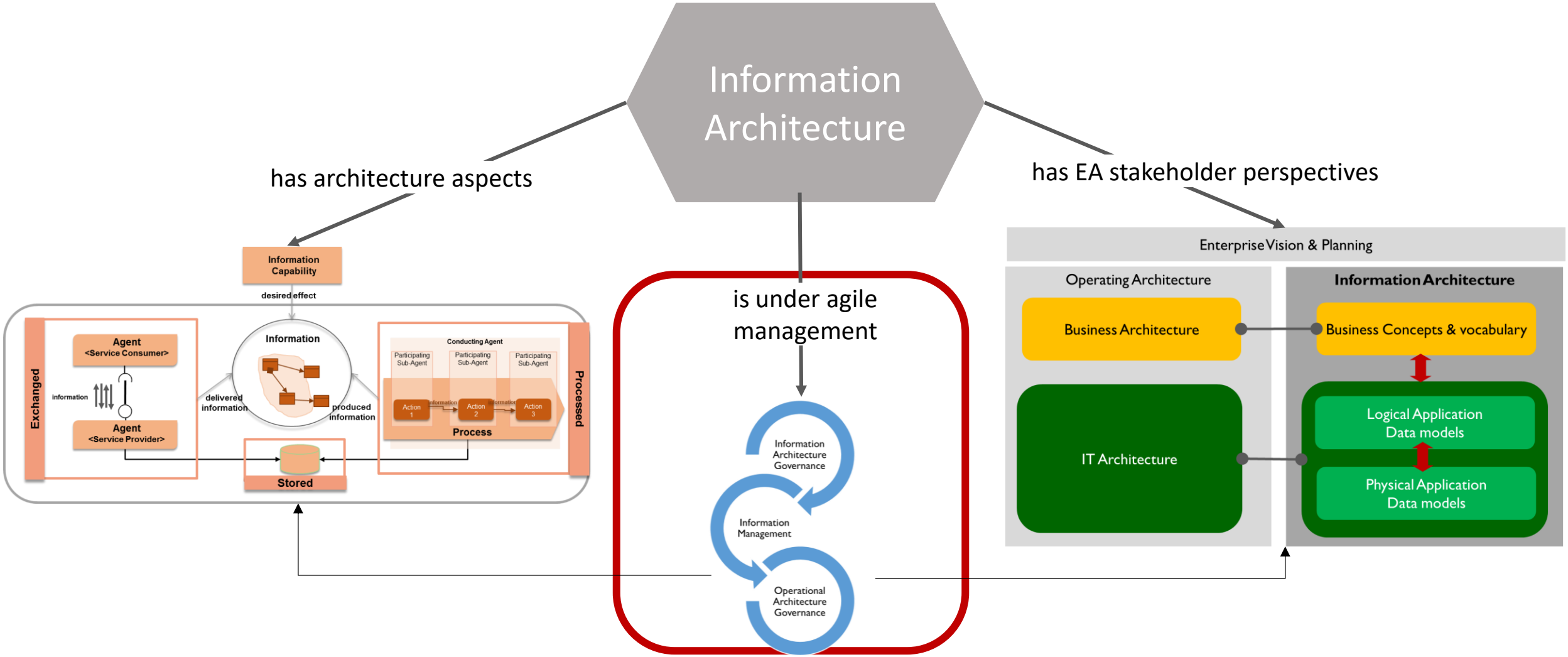

Enterprise
Stage 1
Stage 2
Stage 3
Information Dictionaries
Subject Area
Concept
Concept
Concept
Business Information Domain
Business Information Domain
Logical Data Package
Data Entity 1
Data Entity 2
Data Entity 2
Data Domain
Data Domain
Logical Data Package
Data Entity 1
Data Entity 2
Data Entity 2
Physical Data Domain
Physical Data Domain
selected business information Domain
selected Data Domain
Business Information Map
Business Information Domain 1
Business Information Domain 2
Business Information Domain 3
Business Information Maps provide of mechanism to scope information entities involved in each enterprise stages
Data Domain Map
Data Domain 1
Data Domain 2
Data Domain Map
Data Domain X
Data Domain Y
selected business information domain
Another Business Information Map
Business Information Domain 1
Business Information Domain 2
Business Information Domain 4
selected Data Domain
Data Domain Map
Data Domain 1
Data Domain 3
Data Domain Map
Data Domain X
Data Domain Z
Enterprise transformation


Business Information Domain
Business Information Domain 1
CR
CRUD
R
Business Information Domain 1
CR
CRUD
R
Business Architect
Logical Data Domain
Logical Data Domain 1
CR
CRUD
R
Logical Data Domain 2
CR
R
CRUD
realizes
Physical ER Data Domain
ER Data Domain 1
CR
CRUD
R
ER Data Domain 2
CR
R
CRUD
Data Architect

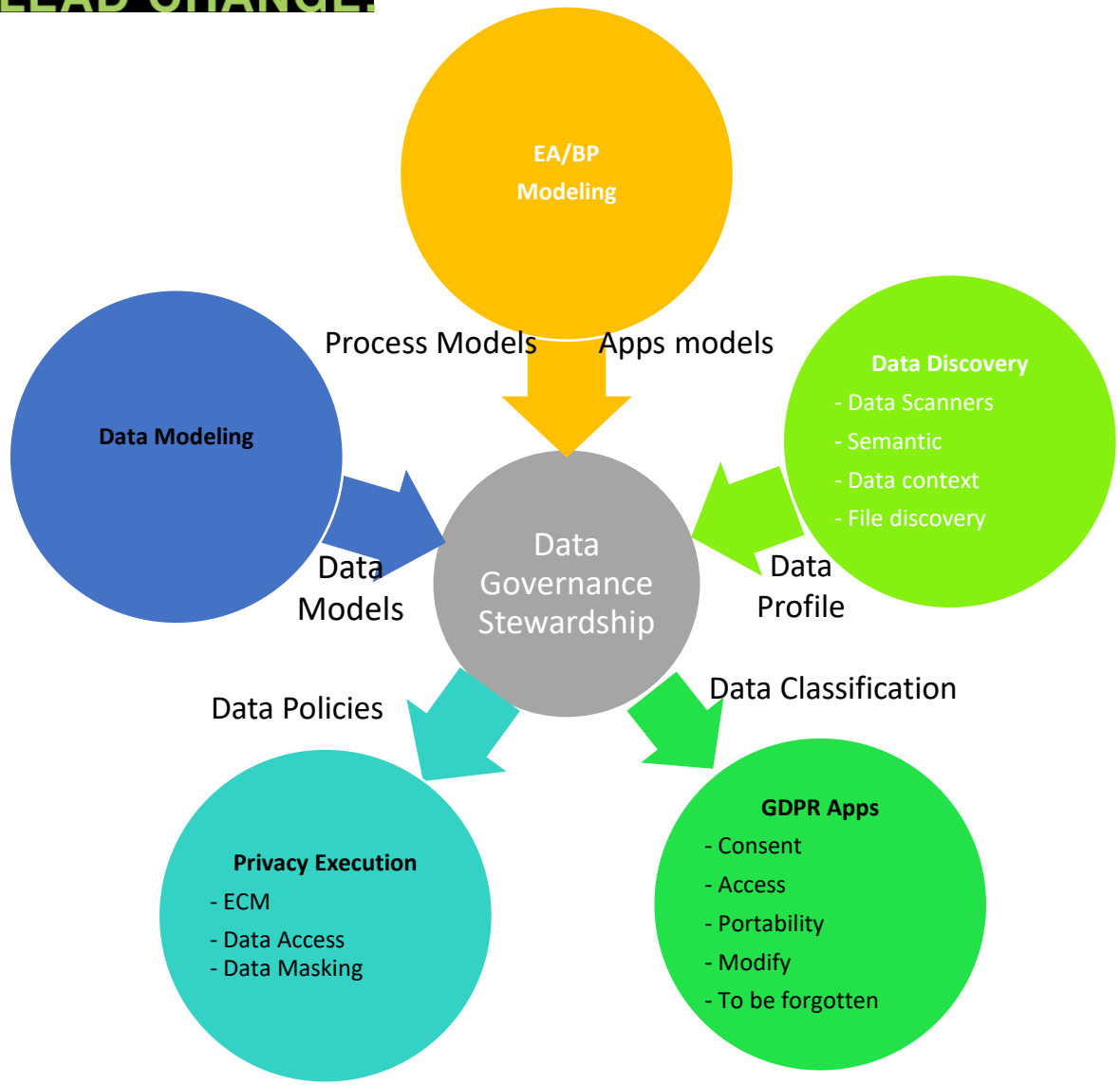
Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

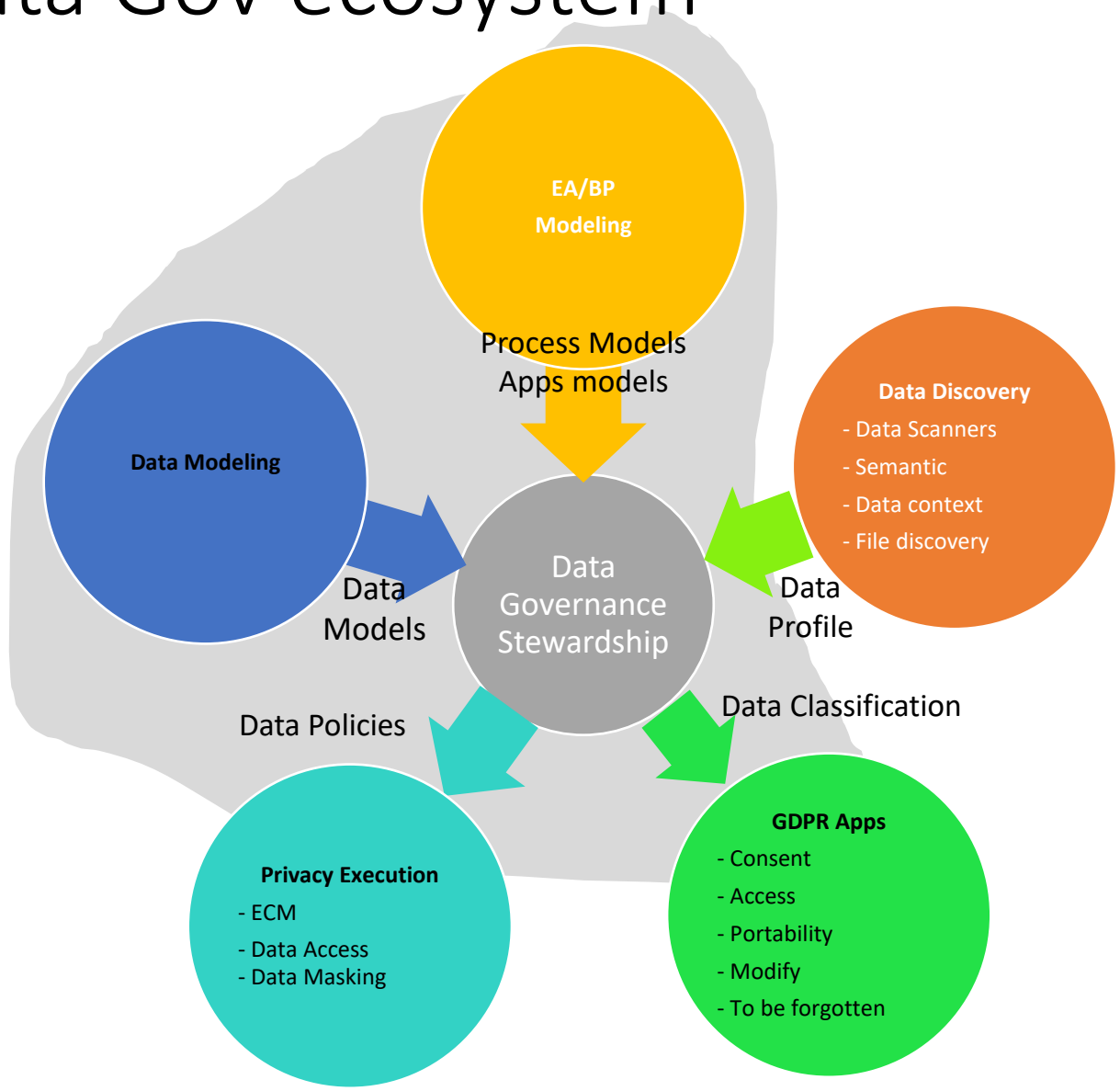
Information Architecture STAKEHOLDER PERSPECTIVES



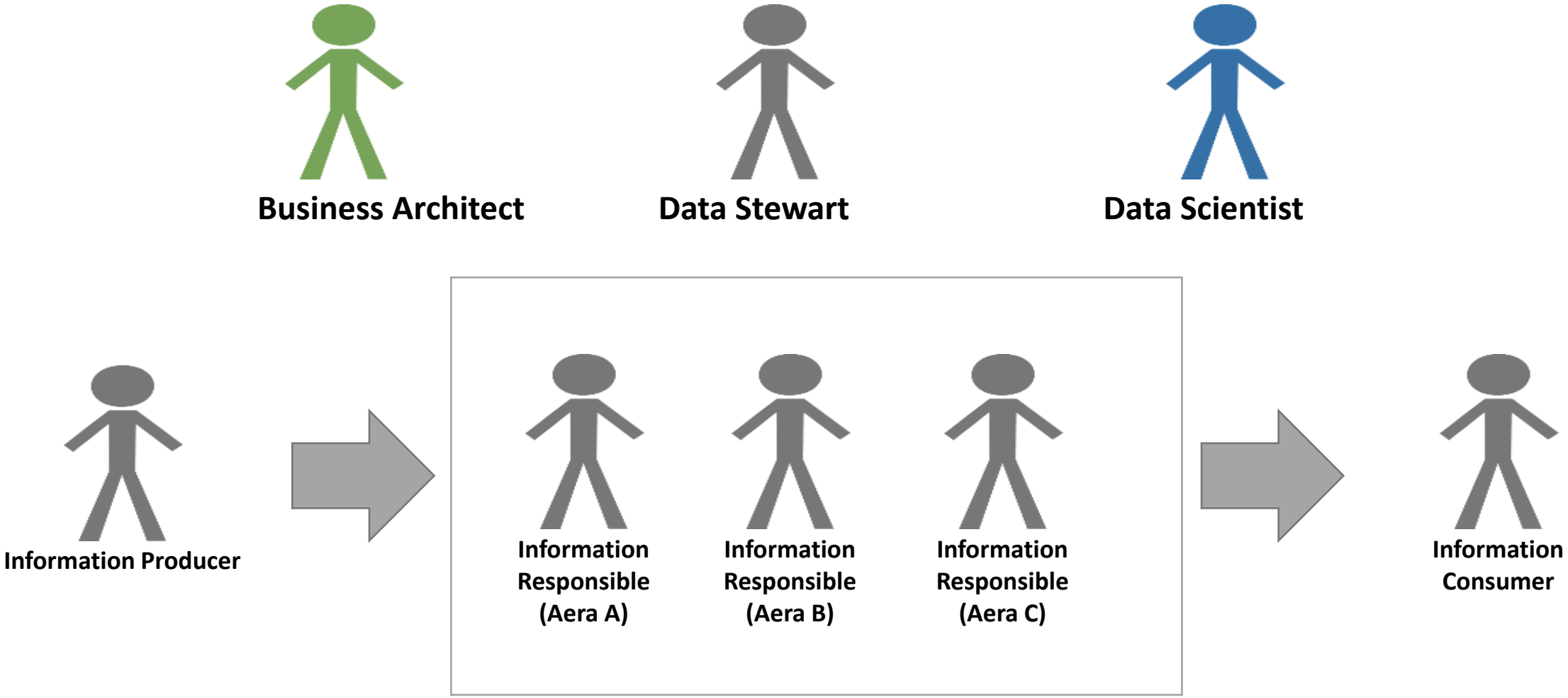
FORRESTER®
CHALLENGE THINKING. LEAD CHANGE.



HOPEX in Data Gov ecosystem



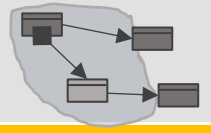
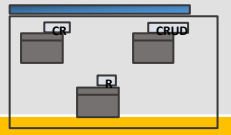
Information architecture is managed



Information Dictionary

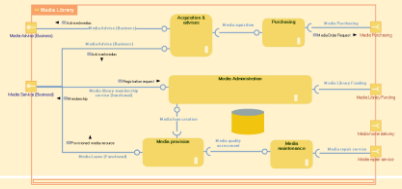
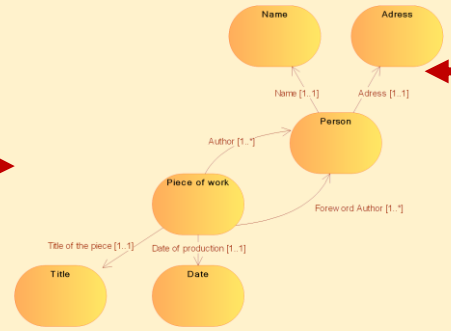
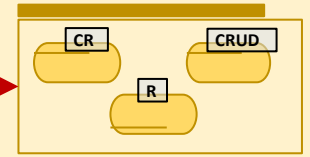
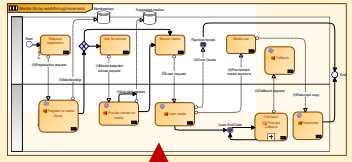
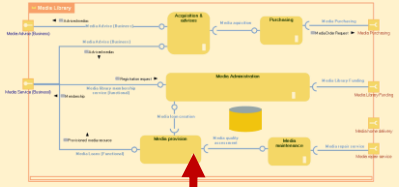
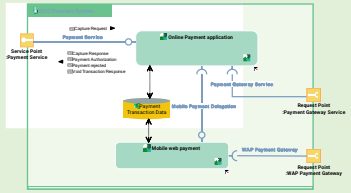
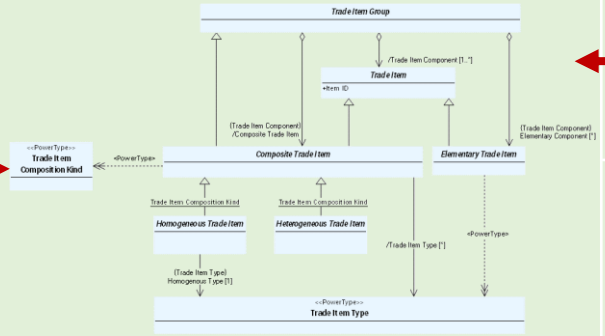
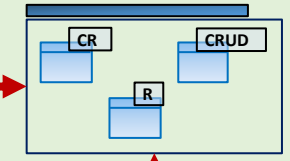
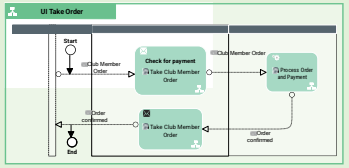
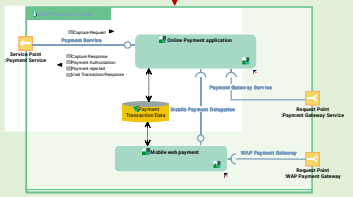
- The business information glossary
- The business information control dictionary
- The internal audit trail on information control

Information Pattern across EA layers

	Information Dictionary	Information Entity	Information Domain	Information Store
				
Business Information	Business Dictionary (Dictionary of concepts)	Concept State/Event	Business Information Domain (was Dictionary Graph)	Business Information Store
Logical Data	Package (Dictionary of Classes)	Class	Logical Data Domain	Logical Data Store
Physical Data	Package (Dictionary of Classes) Database (Dictionary of Tables)	Class Table	ER Physical Data Domain RDB Data Domain	Physical Data Store
Other Physical Data	NoSQL Building Block Dictionary	Record NoSQL Entity	File Structure NoSQL Data Domain	Physical Data Store

- Elements in red will be added in next releases of MEGA.

Data Asset Management Aspects across Layers

	Information Flows	Information Views	Information Reference Model	Information Responsibility Areas
Business Layer	**Business Information Flows**	**Business Information Views**	**Business Information Reference Model**	**Business Information Domain**
	Interactions 	Callback request Base Type Components Loan start Loan end Loaned publication Publication Borrower	Glossary & Terms 	 Grouping
	Process flows 			 Storage
Application Layer	**Application Data Flows**	**Application Data Views**	**Application Data Reference Model**	**Application Data Domain**
	Interactions 	Work data view Base Data Entity Physical Support Author Publisher	Data dictionary 	 Grouping
	Process Flows 			 Storage

Information Assurance

Assurance Analysis Layers		Assurance Domains		
		Information Quality/Privacy Assurance Domain	Process Quality Assurance Domain	Risk & Security Assurance Domain
Policy Architecture		Information Policies & Rules	Quality Policies & Rules	Risk/Security Policies & Rules
Assurance enabled Enterprise Architecture	Capability Architecture	Information Assurance capability/properties	Quality Assurance capability/properties	Risk & Security Assurance capability/properties
	Organizational Architecture	Privacy/Quality Rules Enforcement Organizational Control of Information	Business Rules Enforcement Organizational Control of Quality	Security Business Rules enforcement Risk Control
	System Architecture	System Data Rules System Data Control	System Data Rules System Quality Control	System Data Rules System Security Control
Guide & Trace Operations		Link with Operational Tools	Link with Operational Tools	Link with Operational Tools
Assurance Trustworthiness		Data Quality Assurance	Product/Process Quality Assurance	Risk & Security Assurance

Enhanced EA GRID

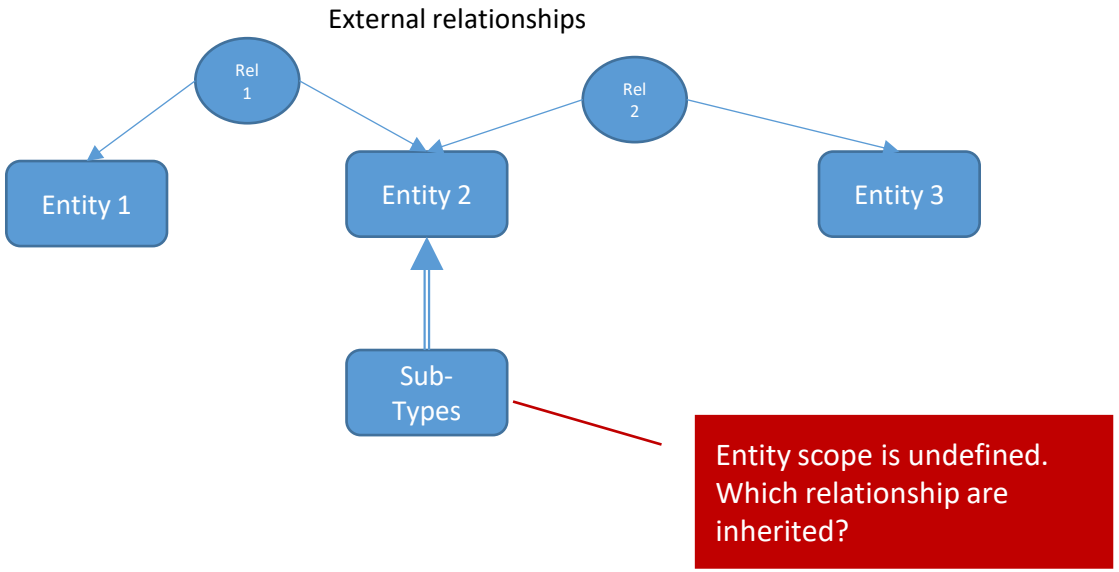
EA Stakeholders	Object Kind → ↓ EA Layer	Enterprise Description Blocks								
		Capability Building Blocks	Outcome	Operating Building Blocks				Information Building Block	Information Container	Policy Building Blocks
				Agent						
				Service [I]	Process	Scenario	Store			
Business Architect Strategic Planner	Business Layer	Business Capability	Business Outcome	Business Functional Area / Business Function				<Business Information> Concept Concept State	Subject Area	Business Rules
				Exchange Contract	Value Stream	Business Scenario	Business Store			
Business Analyst Quality Manager	Organization Layer	Business Capability Skill	Content	Org-Unit				Class Data Entity	Package	Operational Rules
				Exchange Contract	Org-Process	Operational Senario	Logical Data Store			
Application Architect Data Architect Asset Manager	Software Layer	Business Capability Functionality	Content	Application System/Application/IT Service				Class Table	Package Database	System Rules
				Exchange Contract	System Process	Application Scenario	Physical Data Store			
Technical Architect	Technology Layer	Technical Functionality	Communication Format	Technical Applcation Architecture / Software Technology				Table NoSQL Files		System Rules
				Communication Protocol	None	None	Physical Data Store			
Technical Architect	IT Hardware Layer	Functionality	Content	Hardware Artifact				Table NoSQL Files		System Rules
				Exchange Contract	System Process	None	None			
Technical Architect	Hardware Layer	Hardware Functionality	Hadware WorkProduct	Hardware Artifact				None : stores are hosted on IT Servers		System Rules
				Exchange Contract	System Process	None	None			
Infra. Architect SoS Architect	SoS & Infrastructure	Business Capability Functionality Technical Functionality	Content	Resource Architecture				None : stores are hosted on Servers		
				Exchange Contract	System Process	Infrasturcture Scenario	Hardware Store			

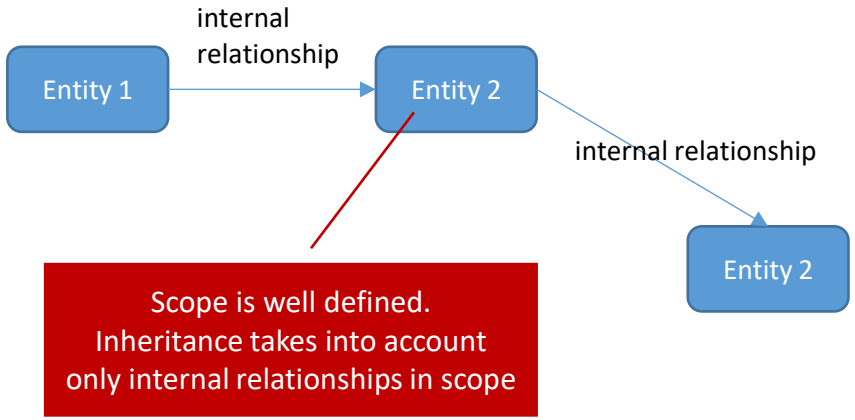
Agenda

- Introduction - Information Architecture Pillars
- Data Asset Management Aspects
- Information Architecture Layers – Stakeholder perspectives
- The Conceptual Layer
- The Application Layer
- The Infrastructure Layer
- Information & Enterprise Planning
- Information Quality Management
- Directed Relationships & Structure

The case of directed relationship

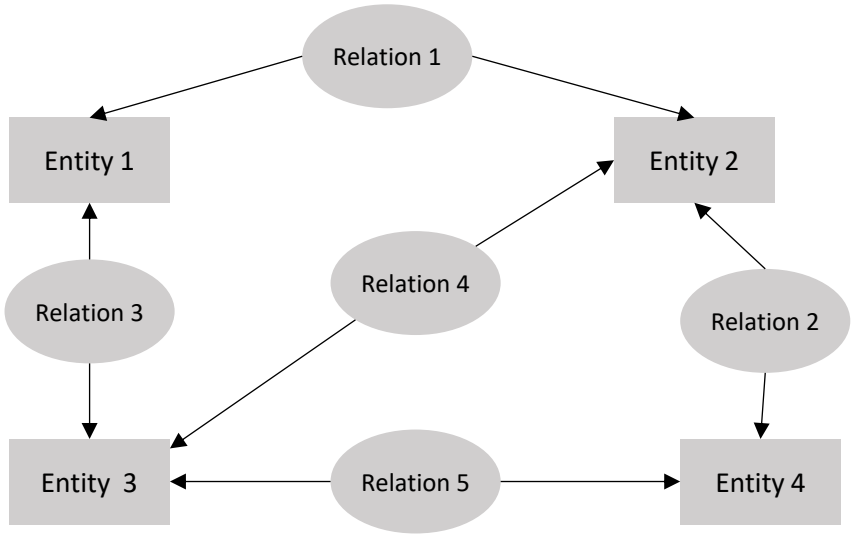
- In the E/R model, Associations and Entities are both nodes in the Data Graph. Hence, Association (relationships) are considered as “external” to entities, preventing from defining a proper “scope” for Entities.
- Internal directed relationships allow to define natural scope of entities.
- UML has both ways to model relationships (parts and associations) and most modern data models are directed graphs.
- Benefits:
  - Natural scope of information entities: what is a “business object?”.
  - Well defined inheritance and redefinition rules coming from OO.
  - Abilities to construct Information Views by navigating on information entity scopes.
  - Ability to construct Information Domains without having to redefine entity scope.

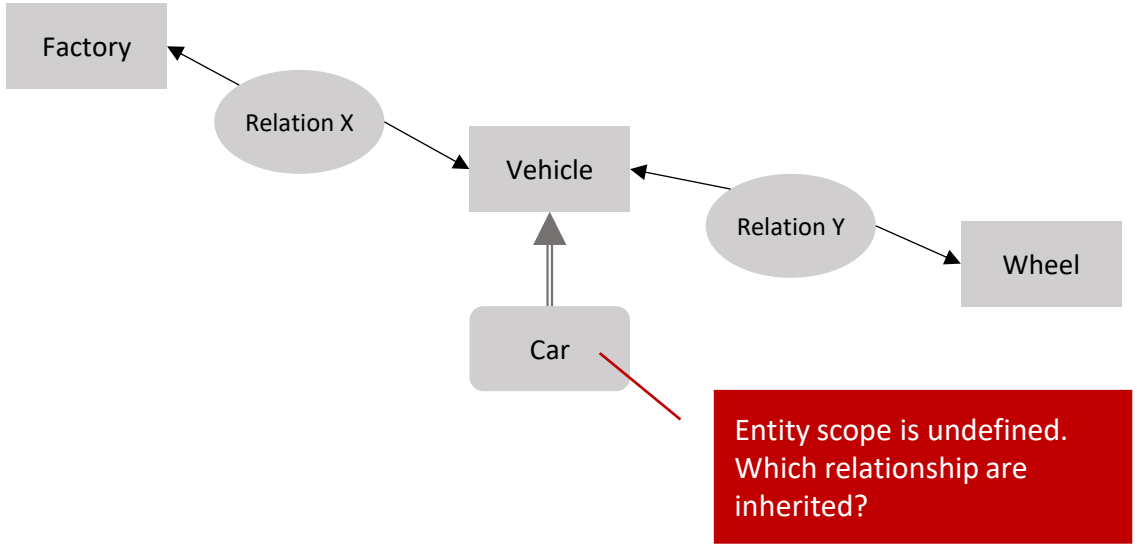




The case of directed relationships

- In the E/R model, Associations and Entities are both nodes in the Data Graph.
- Associations (relationships) are considered as “external” to entities:
  - preventing from defining a proper “scope” for Entities.
  - making inheritance between Entities difficult to handle: which association is in the inheritance scope?





Eric Evans - AGGREGATE – A definition

- The notion of “building block” is well know in our industry. People from the “Domain Driven Design” field call them “Aggregate”.
- Below is the definition of Aggregate is given by Eric Evans who coined the term “DDD: domain driven design”:
  - *First we need an abstraction for encapsulating references within the model. An AGGREGATE is a cluster of associated objects that we treat as a unit for the purpose of data changes. Each AGGREGATE has a root and a boundary. The boundary defines what is inside the AGGREGATE. The root is a single specific ENTITY contained in the AGGREGATE. The root is the only member of the AGGREGATE that outside objects are allowed to hold references to, although objects within the boundary may hold references to each other. ENTITIES other than the root have local identity, but it only needs to be unique within the aggregate, since no outside object can ever see it out of the context of the root ENTITY.*

Eric Evans - AGGREGATE – Example 1

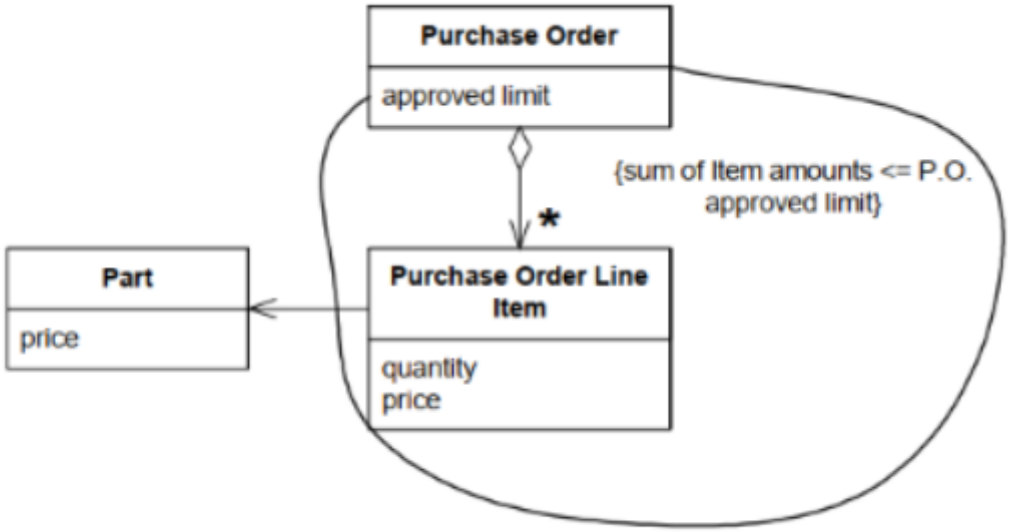
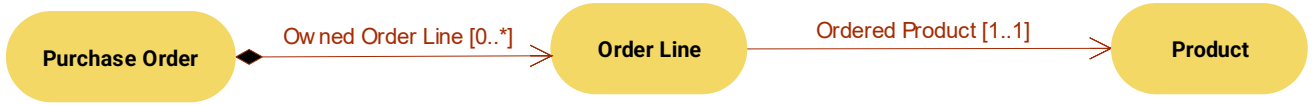


Figure 6. 5

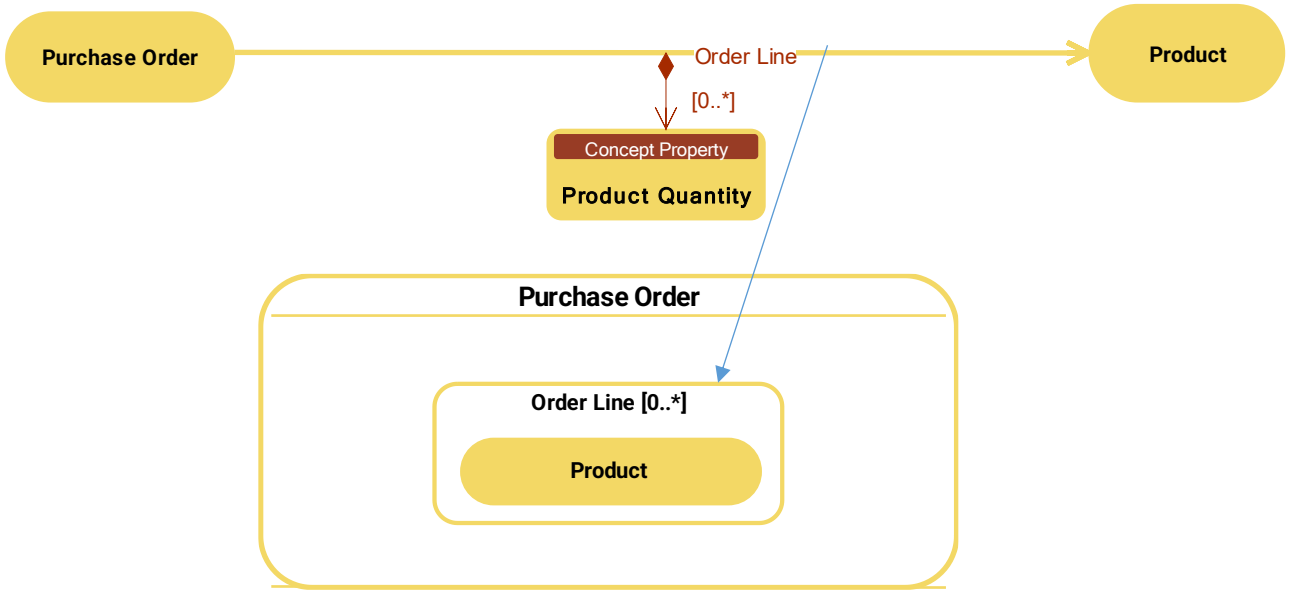
Page 93:
Eric Evans - Domain Driven Design - Reformatted version.pdf

Structure Sample

- In traditional E/R, “order line” is a concept of the same nature as “purchase order”



  - When adopting a structured approach, “orde



AGGREGATE.

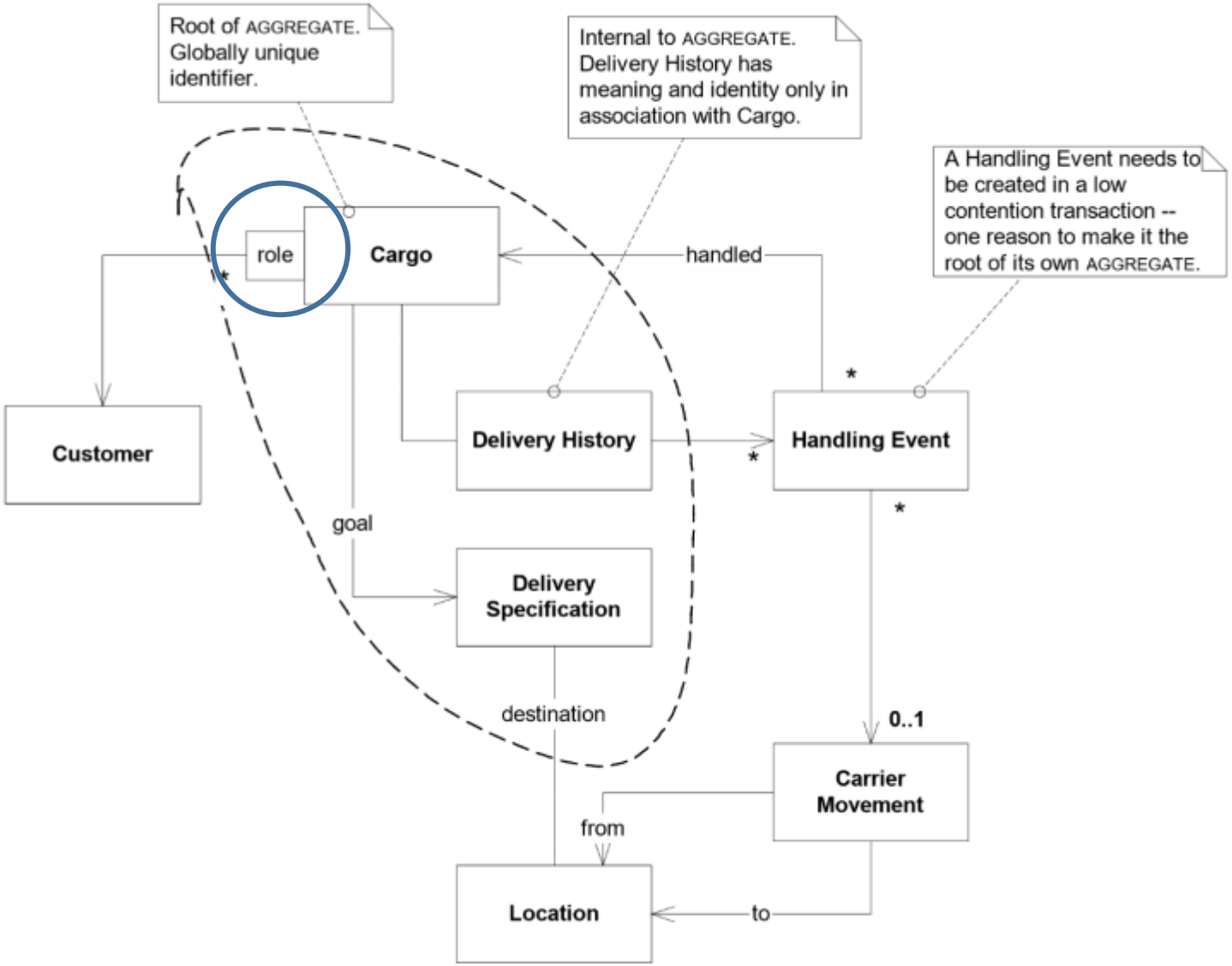


Figure 7. 3

Eric Evans - AGGREGATE – Example 2

Cluster the ENTITIES and VALUE OBJECTS into "AGGREGATES" and define boundaries around each. Choose one ENTITY to be the "root" of each AGGREGATE, and control all access to the objects inside the boundary through the root. Only allow references to the root to be held by external objects. Transient references to internal members can be passed out for use within a single operation only. Because the root controls access it cannot be blind-sided by changes to the internals. This makes it practical to enforce all invariants for objects in the AGGREGATE and for the AGGREGATE as a whole in any state-change.

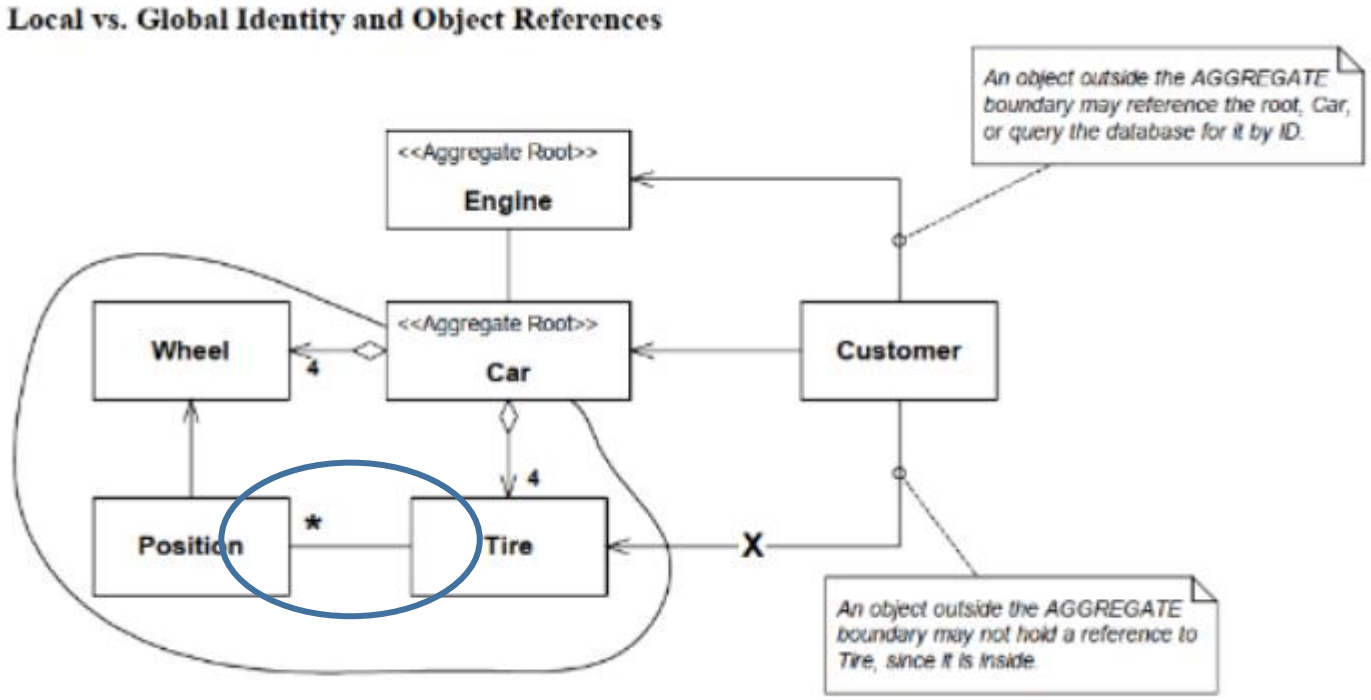
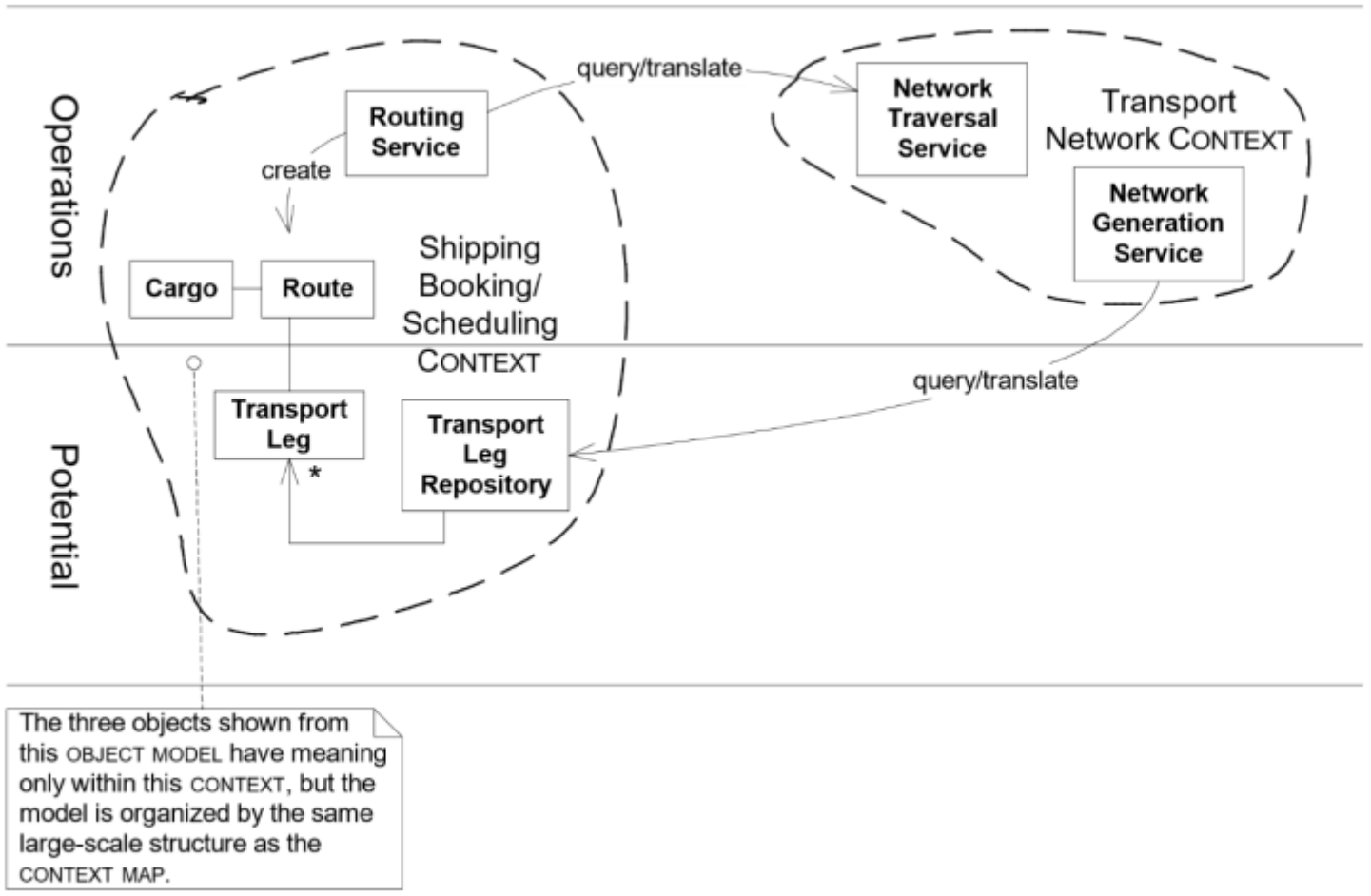


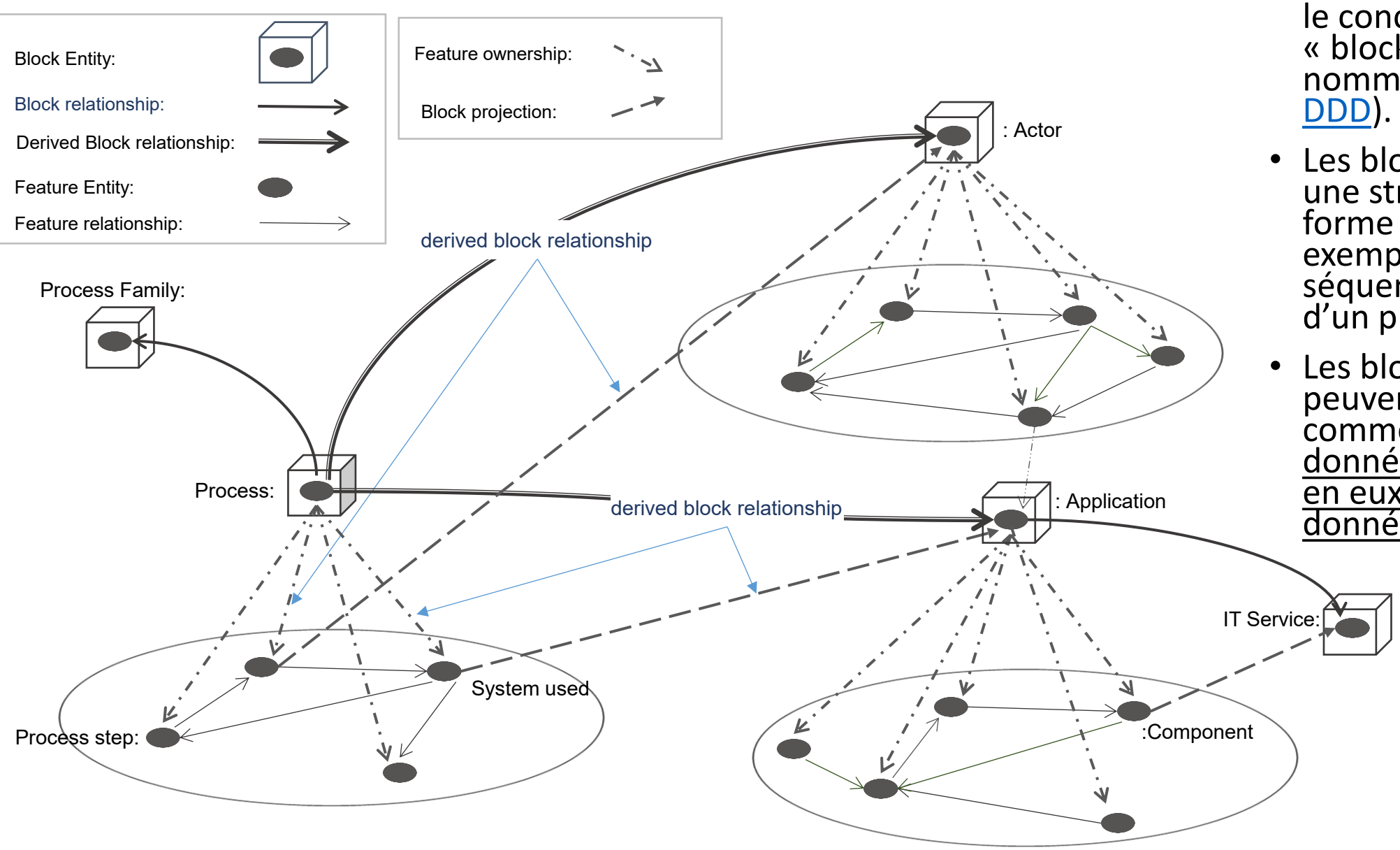
Figure 6. 2

Eric Evans - AGGREGATE – Example 3



Page 328

Le débat: vues versus blocks



- Le graphe primaire doit être étendu pour intégrer le concept de « block structuré» aussi nommé « Aggrégat » (cf le DDD).
- Les blocks structurés ont une structure interne qui forme un sous-graphe (par exemple, les opérations, séquences et participants d'un processus).
- Les blocks structurés peuvent être parcourus comme des vues de données mais ne sont pas en eux-mêmes des vues de données.

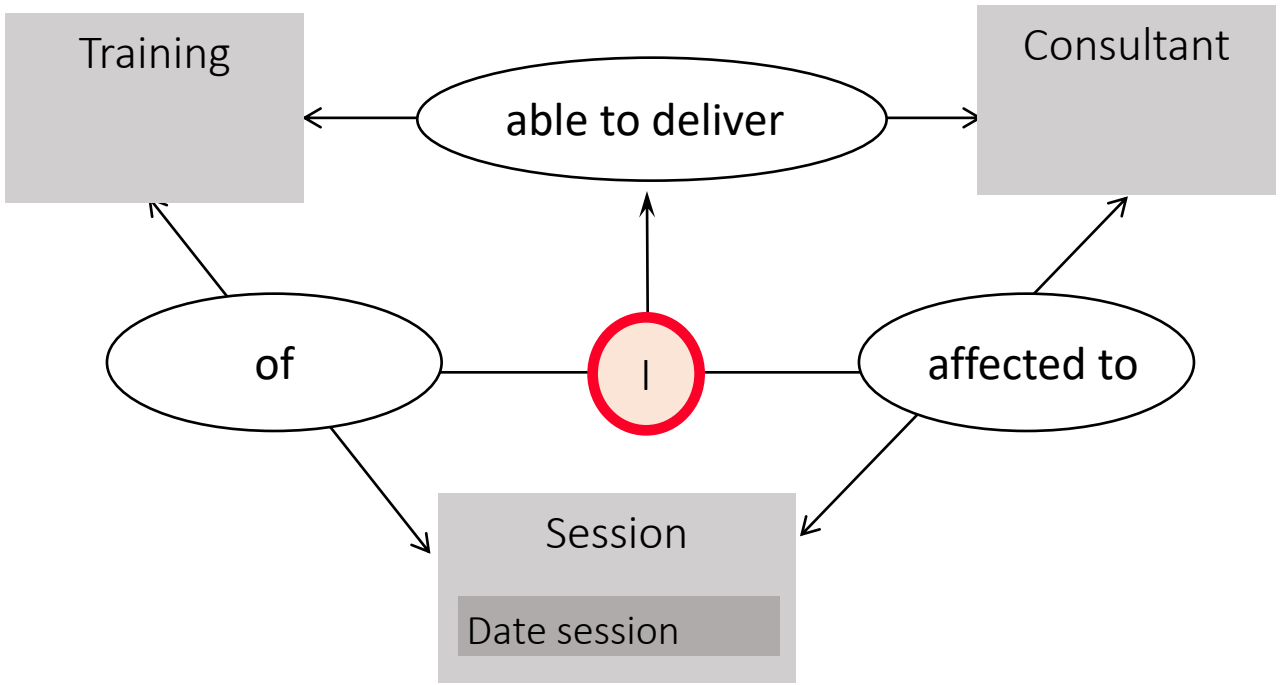
Data Modeling - Lessons learned

- Entity types should have a local identifier within a database or exchange file. These should be artificial and managed to be unique.
- Entity types should represent, and be named after, the underlying nature of an object, not the role it plays in a particular context.
- Entity types should be part of a subtype/supertype hierarchy (class hierarchy) in order to define a universal context for the model.
- Activities and associations should be represented by entity types (not relationships or attributes).
- Relationships (in the entity/relationship sense) cannot be referred to directly as objects, so should only be used to represent things that you do not need to refer to independently, such as the involvement of something in an activity or relationship.
- Candidate attributes should be suspected of representing relationships to other entity types.
- The first of these principles makes clear the importance of “the nature of things” that is at the heart of ontology. A data model is an ontology, and as such makes ontological commitments, though these are rarely explicitly acknowledged.

- Source : Matthew West
  - Matthew West - Publications (matthew-west.org.uk)

Entities & external relationships - Example

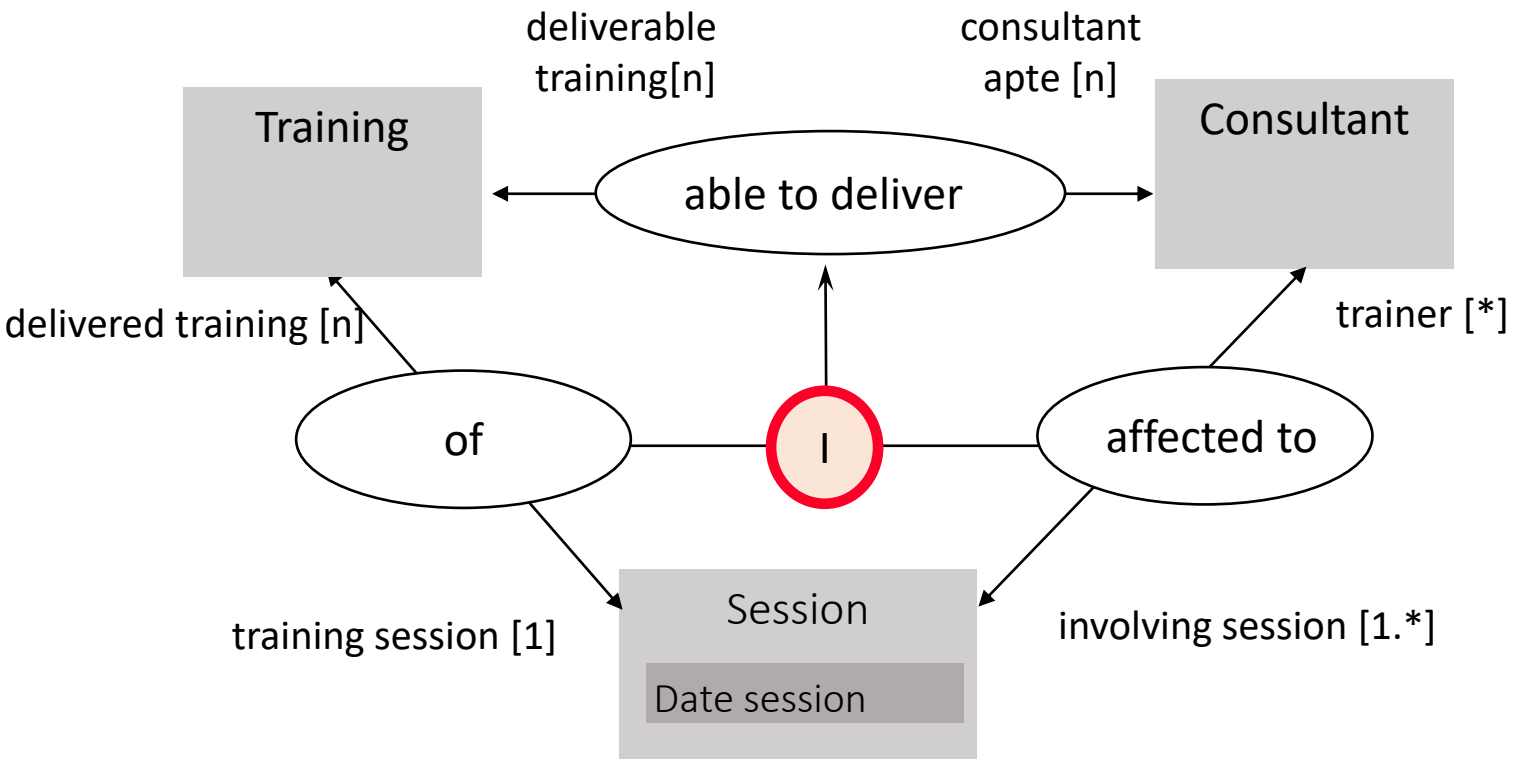
- Consultants can provide trainings
- Trainings are delivered offering sessions, at a particular date, by consultants



External relationships and roles - à la Merise

- Association Ends (Roles) provide a better readability of associations …but .. they do not help in defining the scope of object definitions.

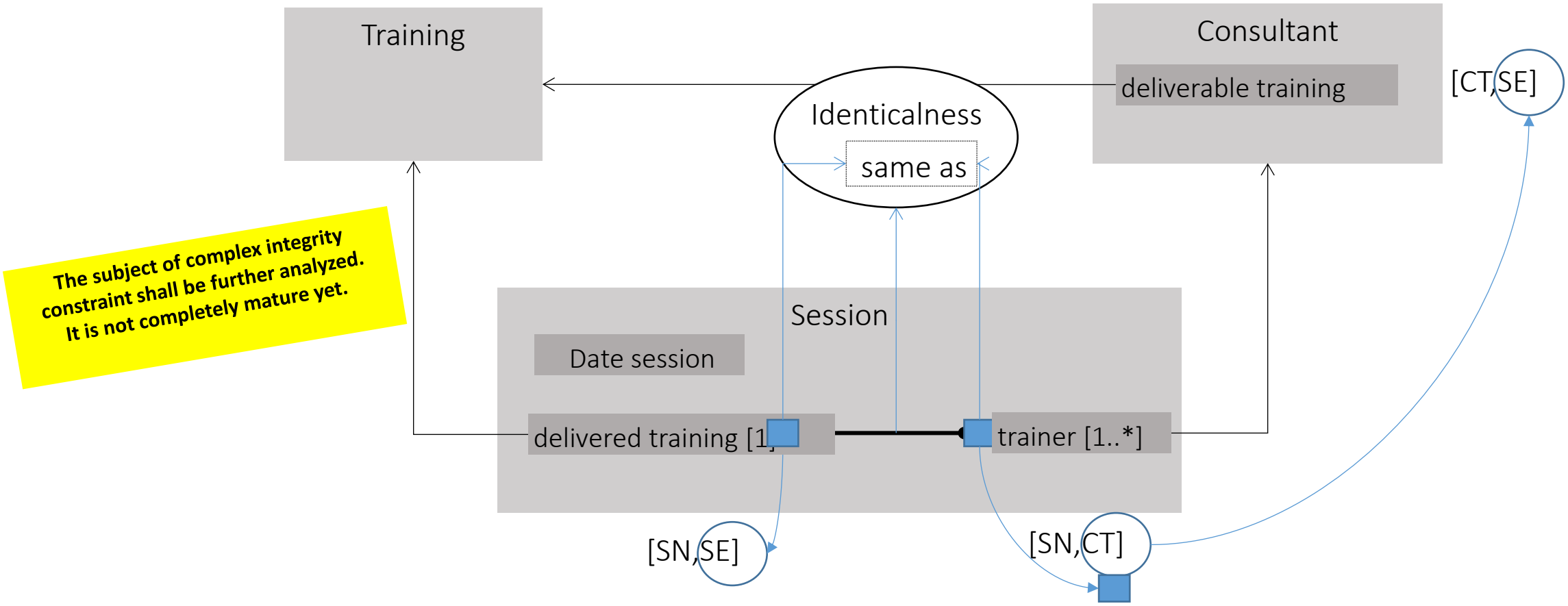
- Does “trainer” belongs to “Consultant” or “involved training” to Session?



From external to internal relationships

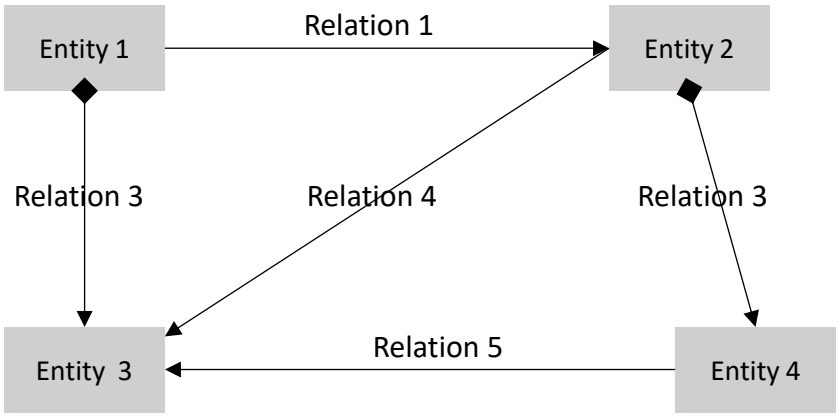
- Roles are played within composite structures

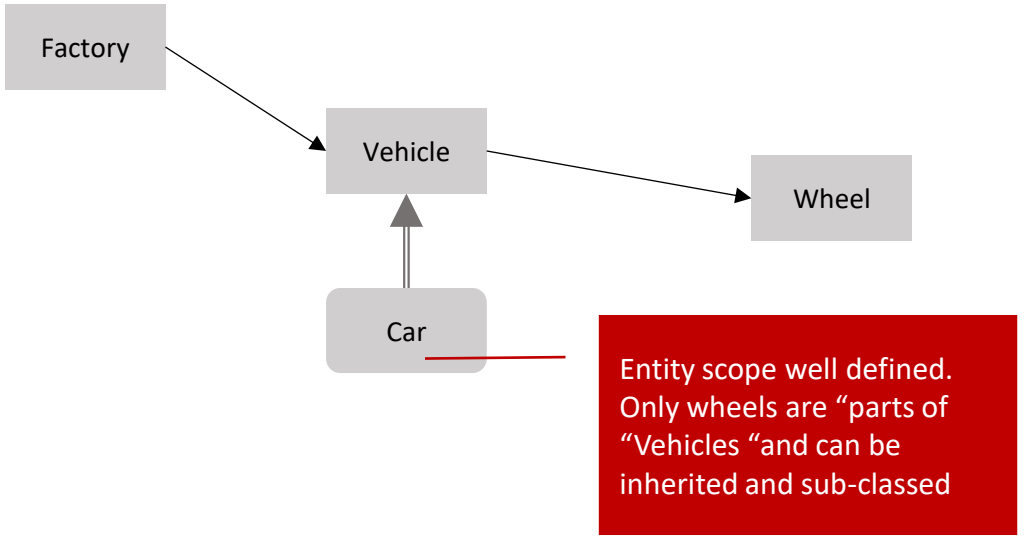
- Relationships are between roles in composite structures



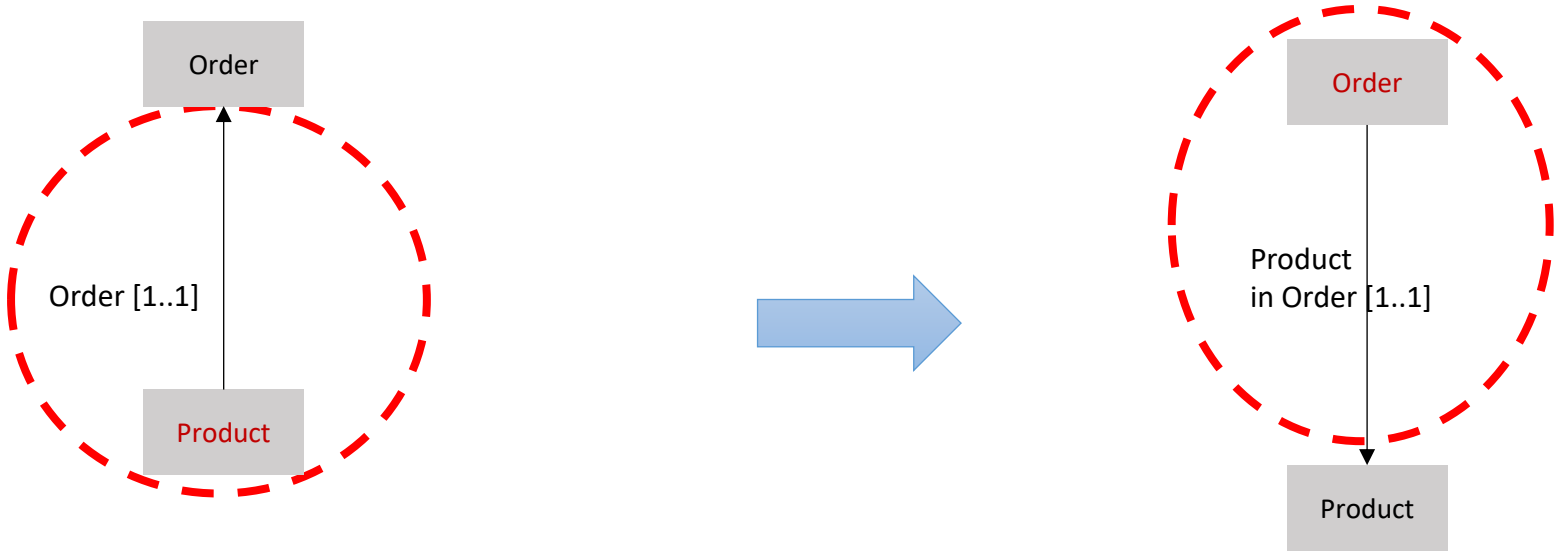
Whole/Parts and Directed Graph

- In a whole/part models, relationships are edge in the graph; only Entities are nodes.
- Entities scope are directly given by following relationship directions.





Whole/part relationships: Relationship direction rule

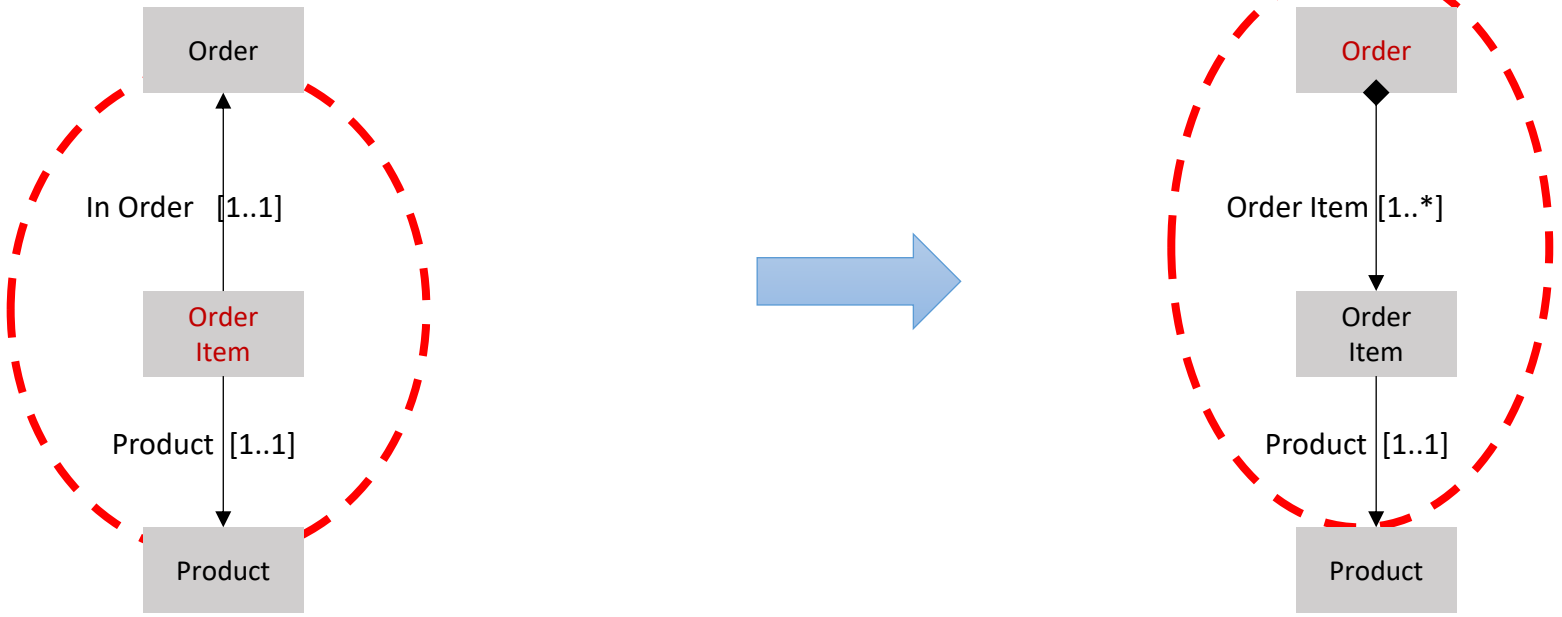


Products are not made of Orders…

… but Orders are always about Products.

Tip : Finding the right direction might seem tough at the beginning. A good way to know is to wonder *“Who would loose information if I cut the link ?”,* or *“Who knows the other ?”*

Whole/part relationships: Relationship direction rule



Order Items are not made of Orders…

… but Orders are always about Order Items.

Tip : Order Items are specific kind of entities that cannot exist without their whole. It doesn't mean they are defines by their whole. The whole/part relationship between whole and such entities is a composition relationship (black diamond).

The rule for finding the right direction steel apply: *"Who would loose information if I cut the link ?".* Without Order Items, Orders are undefined.

Principles

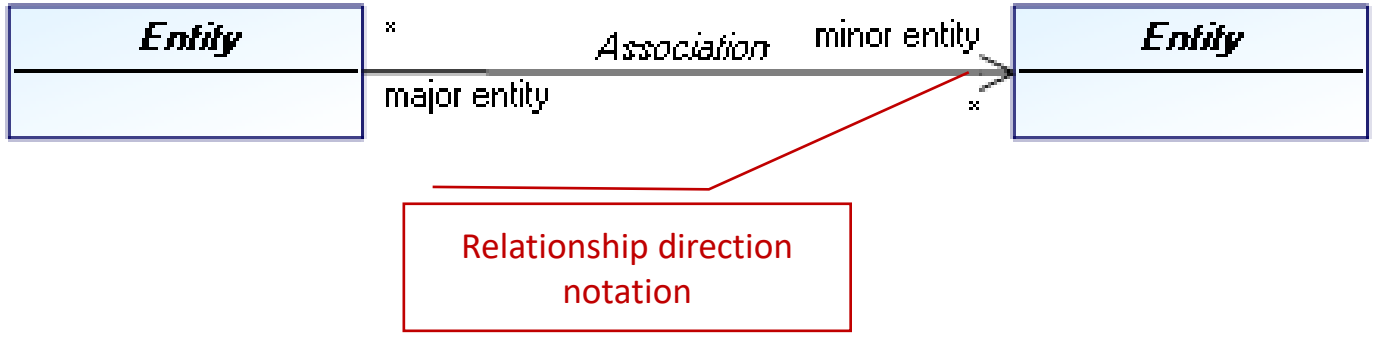
- No use of attributes
- Relationships are all directed
- Sub-typing provide a semantic integration framework
- Redefinition is a tool for managing variability
- Roles provide views of entity types
- Powertype provide classification of entity types.
- Multiple terminology can apply at the conceptual level.
- Occurrences provide validation schemes
- Structural relationships

Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

Relationship directions (1)

- A data model is a directed graph which means that all relations shall be directed.
- Relation direction indicates that “source entities” are “relation owners” which means that these relations are part of the definition of “source entities”.
- Proposed graphical notation: use an open arrow to represent Relationship directions

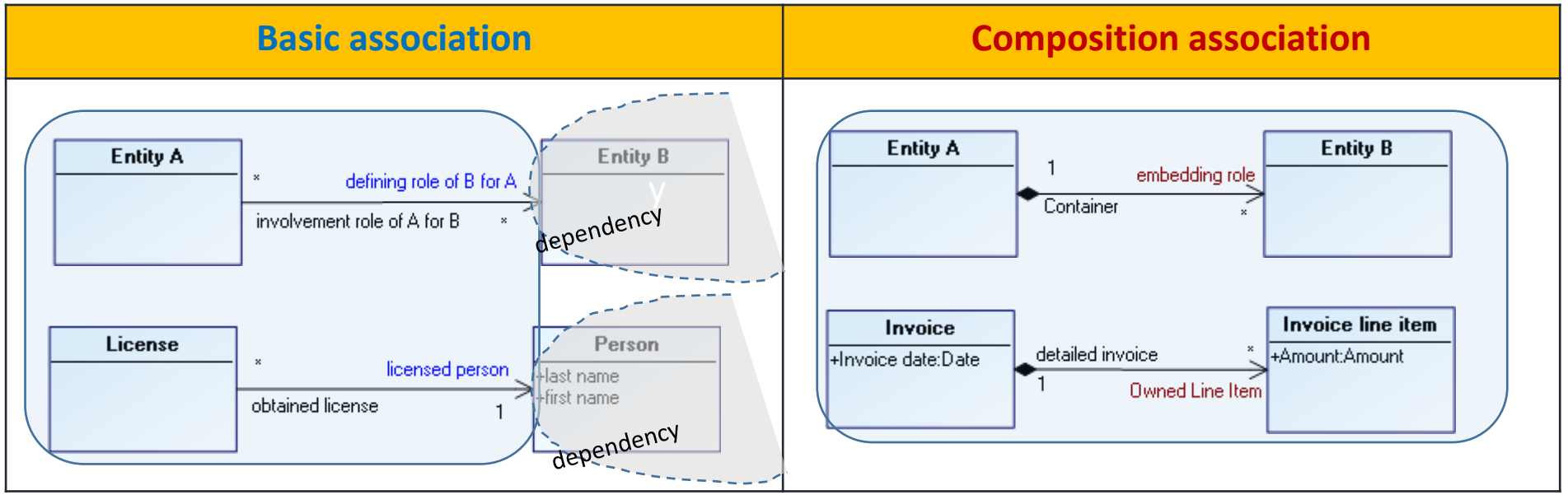


Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

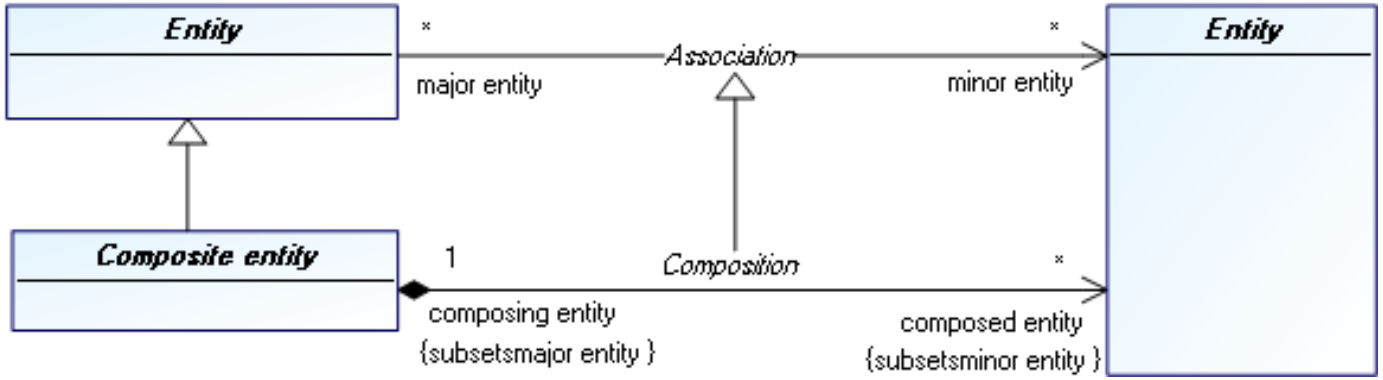
Association & object scope

- Object scopes are defined according to the nature of associations they have with other entities:
  - Basic association : target occurrences are merely referenced by source occurrences. Basic associations express dependencies between object scopes.
  - Composition association: the existence of target occurrences is tied to the existence source occurrences.



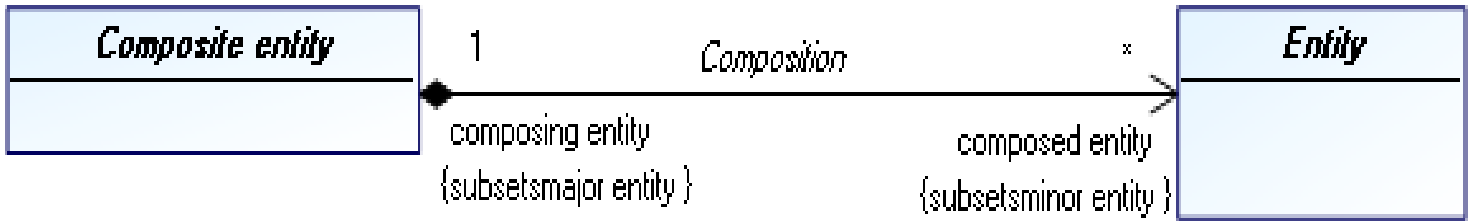
Composition associations (1)

- Composition associations are kinds of associations that define ownership relationships between composite entities and other entities.



Composition associations (2)

- Notation
  - Composition associations use the UML black diamond on the major ‘composing entity’ end.
  - Direction notation could be kept or remove
    - The current proposal is to always show the direction arrow

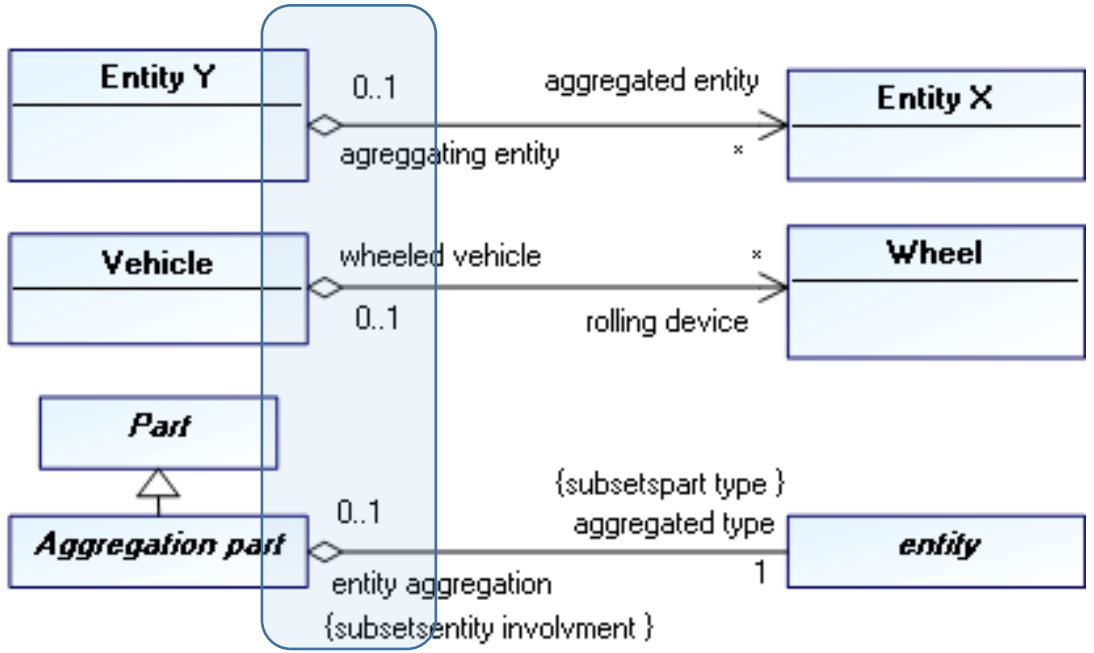


Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

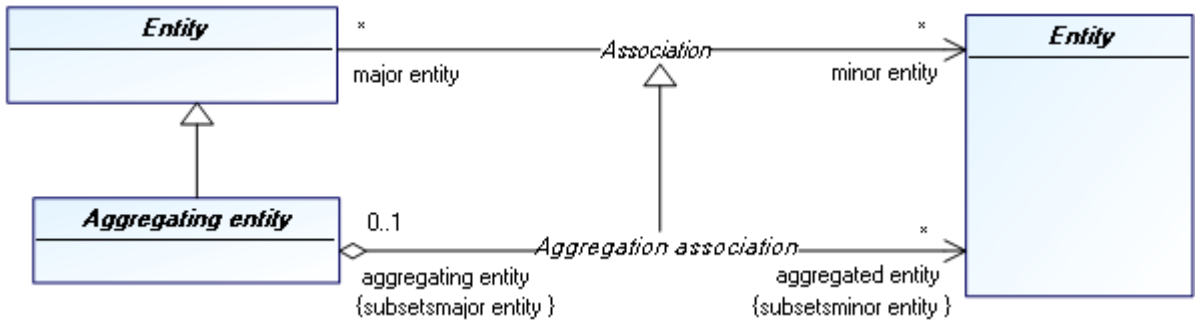
Object usage (1)

- Beyond object scope, the existence of objects may depends on their involvement in associations:
  - Aggregation associations are basic associations where occurrences can be involved at most once with their aggregating occurrence (lower multiplicity=0, upper multiplicity = 1).
  - Reuse actions shall ensure that aggregated entity



Object usage (2)

- ‘Aggregation association’ is a sub-type of ‘Association”.

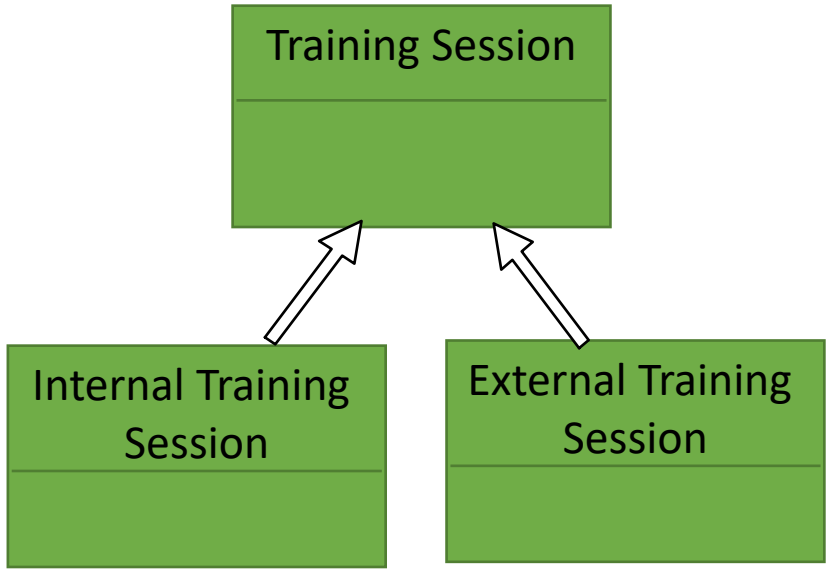


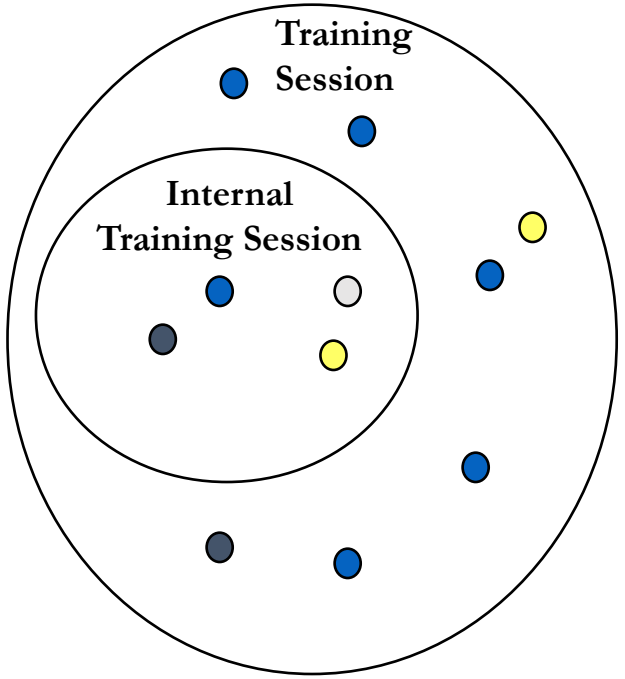
- Notation
  - Aggregation associations use the UML black diamond on the major ‘aggregating entity’ end.
  - Direction notation could be kept or remove
    - The current proposal is to always show the direction arrow

Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

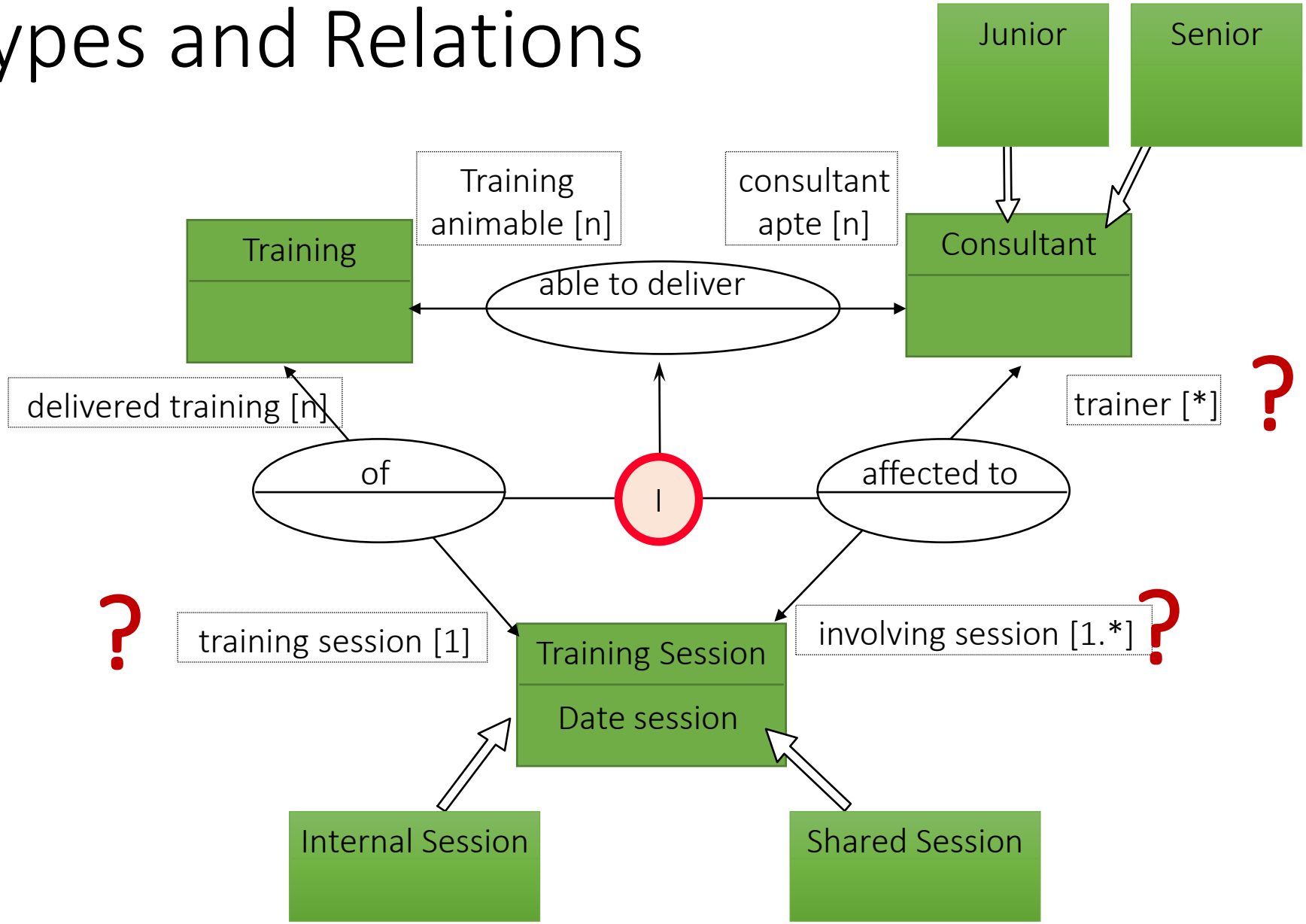
Sub-types & redefinitions





- Internal-Training is a sub-type of Training means that all instances of Internal-Training are instances of Training
- But what about relationships?

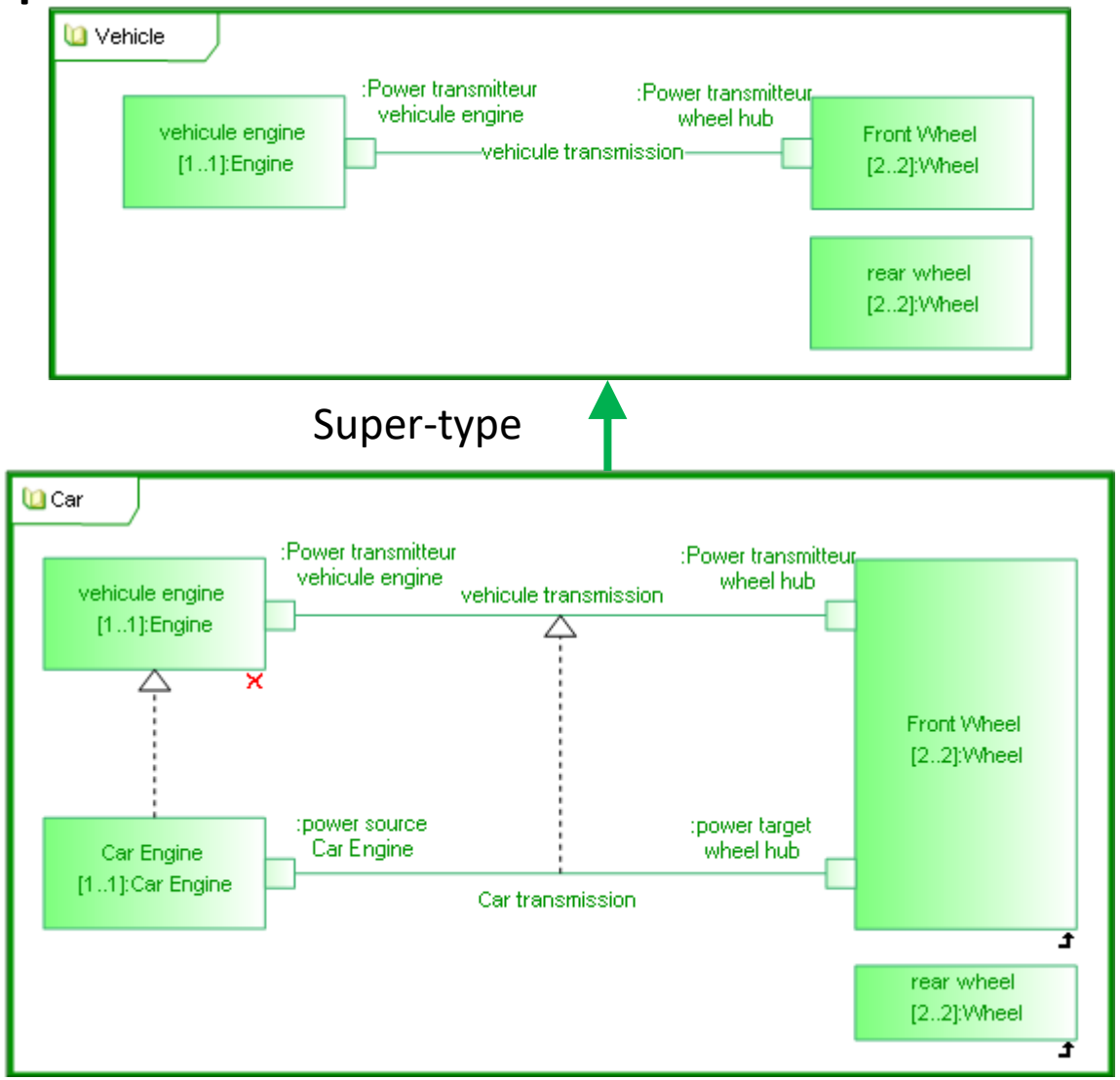
Sub-Types and Relations



Sub-Types and composite structures

- Proper inheritance of “parts” can be managed

- Sub-typing and substitution can be applied to :
  - Parts
  - Relationships



Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

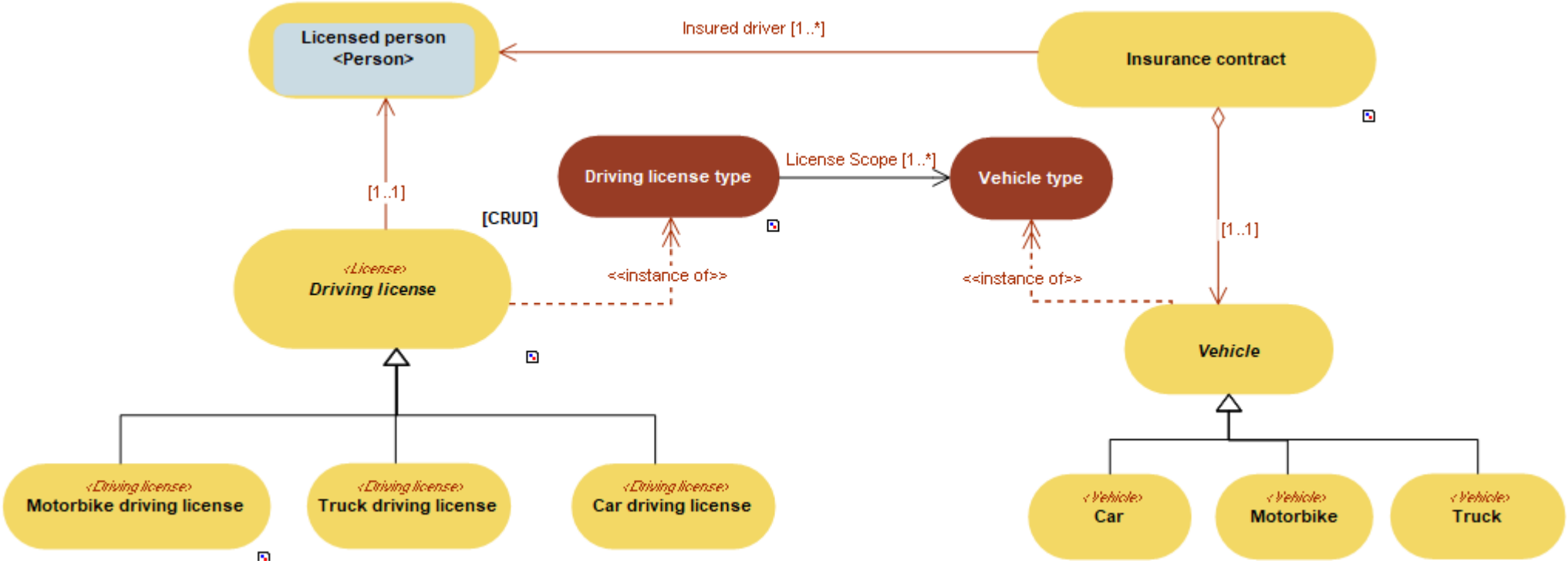
Power Type

- Requirement
  - Many data models have to deal with “kinds of ..” :
    - Kind of contract, kind of vehicle, kind of organization.
  - There is a need to have a unified way to relate data entities these different “kinds of …”.

- Principles
  - Object categories (kinds of ..) are based on an organization of sub-types called power set.
  - A power type is the type of all the sub-type of an entity.
    - Vehicle : Powerset = Car, Motorbike, Truck, …

- Foundations
  - This concept has strong mathematical foundation and is an integral par of set theory :
    - In mathematics, the **power set** (or **powerset**) of any set *S*, written , $P(S)$, $\wp(S)$ or 2^S, is the set of all subsets of *S*, including the empty set and S itself.

Power Type example



- All sub-types of “Driving license” are instances of “Driving license type”
  - => “Driving license type” is a class of class (power type)
- All sub-types of “Vehicle” are instances of “Vehicle Type”
  - => “Vehicle Type” is class of class (power type)

Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Roles
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

A long standing problem : Merise – EROS

Chapitre 16

Spécification des modèles externes et des contraintes : le discours sur les occurrences et l'approche EROS[1]

Au-delà de Merise
Yves Tabourier
1986

16.1. UN SUPPORT VISUEL EXPLICATIF : LA STRUCTURE D'OCCURRENCES

Pour présenter cet exposé je m'appuierai sur l'exemple des contrats d'assurance auto déjà présenté plus haut (chap. 7, fig. 9b, chap. 12 fig. 6b).

Supposons qu'un acteur soit intéressé au lien qui unit les personnes qui souscrivent des contrats à celles qui sont autorisées à conduire sur ces contrats. Pour exprimer ce lien, étant entendu que les rôles de souscripteur et de conducteur sont joués par des personnes, tout support entité/relation possible, quel que soit par ailleurs le mode d'expression (logique formelle, français, etc.) est semblable à celui de la fig. 1.

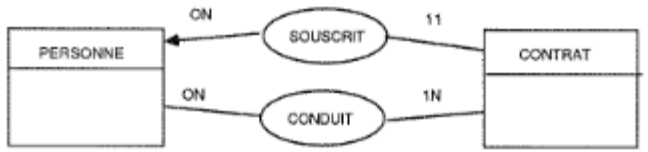


Figure 1. — *Support entité-relation pour exprimer le lien entre souscripteur et conducteur*

1. EROS : « Entity, Relationship, Occurrences Structure ».

Un tel diagramme permet de montrer séparément les personnes jouant les deux rôles. Il reste typologique, en un sens, puisque de nombreuses occurrences de personnes peuvent s'inscrire dans l'une ou l'autre des cases PERSONNE-1 et PERSONNE-2. Cependant, on pourra parler d'une « occurrence » de cette structure, comme définie par :

— une occurrence x de PERSONNE, dans la case PERSONNE-1,
— une occurrence y de CONTRAT, dans la case CONTRAT-1,
— une occurrence z de PERSONNE, dans la case PERSONNE-2.

telles que y soit souscrit par x et prévoie z parmi les personnes autorisées à conduire le véhicule assuré.

Vouloir faire la même chose avec la figure 1 conduirait à placer x et z dans la case « PERSONNE » de ce diagramme.

Si un autre acteur est intéressé par le lien entre les contrats souscrits par une même personne, ou admettant une même personne comme conducteur, il n'aura toujours comme support entité-relation qu'un extrait du diagramme de la figure 1, alors qu'il pourra disposer de deux « structures d'occurrences » spécifiques pour supporter ses deux nouveaux discours (fig. 3).

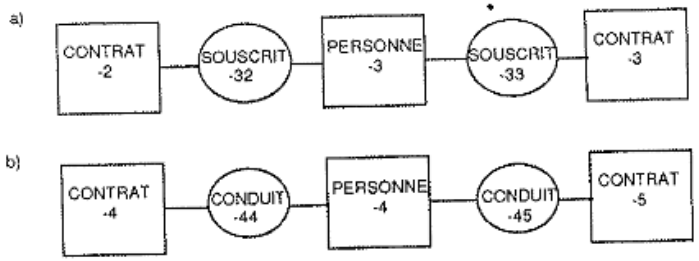


Figure 3. — *Structures explicatives des liens entre contrats*
a) De même souscripteur.
b) Admettant un même conducteur.

Contract example (Merise – EROS)

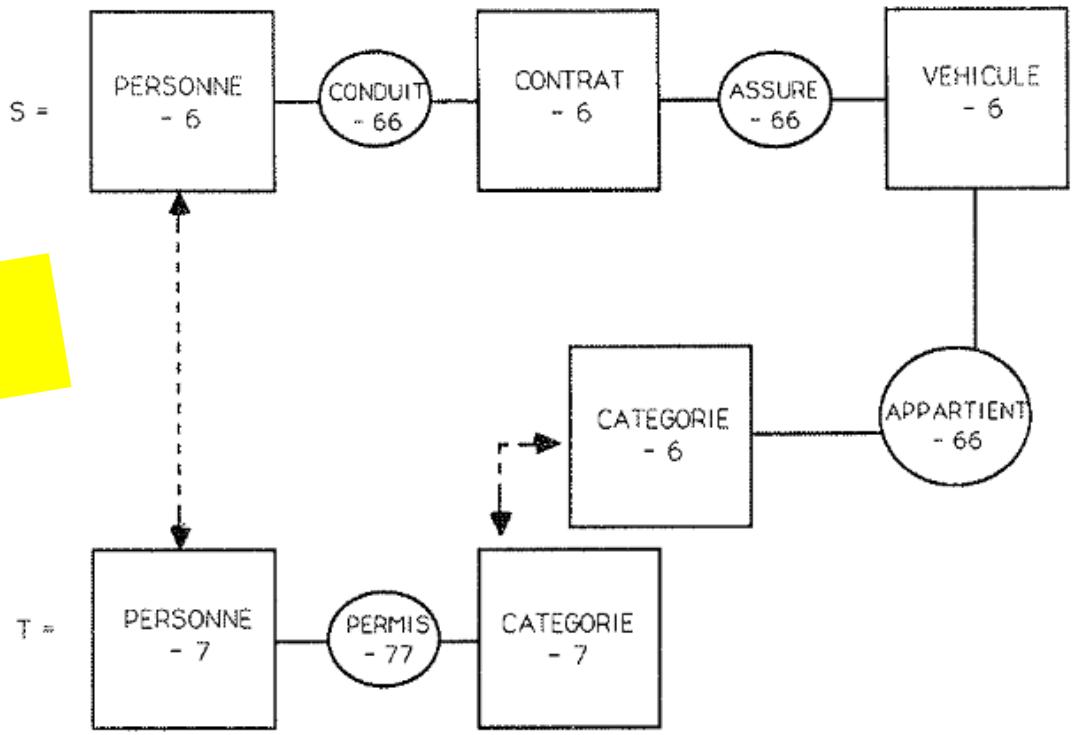
Un exemple tiré du cas de la compagnie d'asurrances montrera l'usage de la notion de correspondance. Si l'on veut dire que « tout conducteur d'un contrat assurant un véhicule d'une catégorie doit avoir un permis pour cette catégorie », on introduira les structures S et T de la figure 7 et l'on écrira :

— avec la primitive « IMPL » (T/N 81) :

IMPL (S ⇒ T/PERSONNE-6 = PERSONNE-7, CATÉGORIE-6 = CATÉGORIE-7)

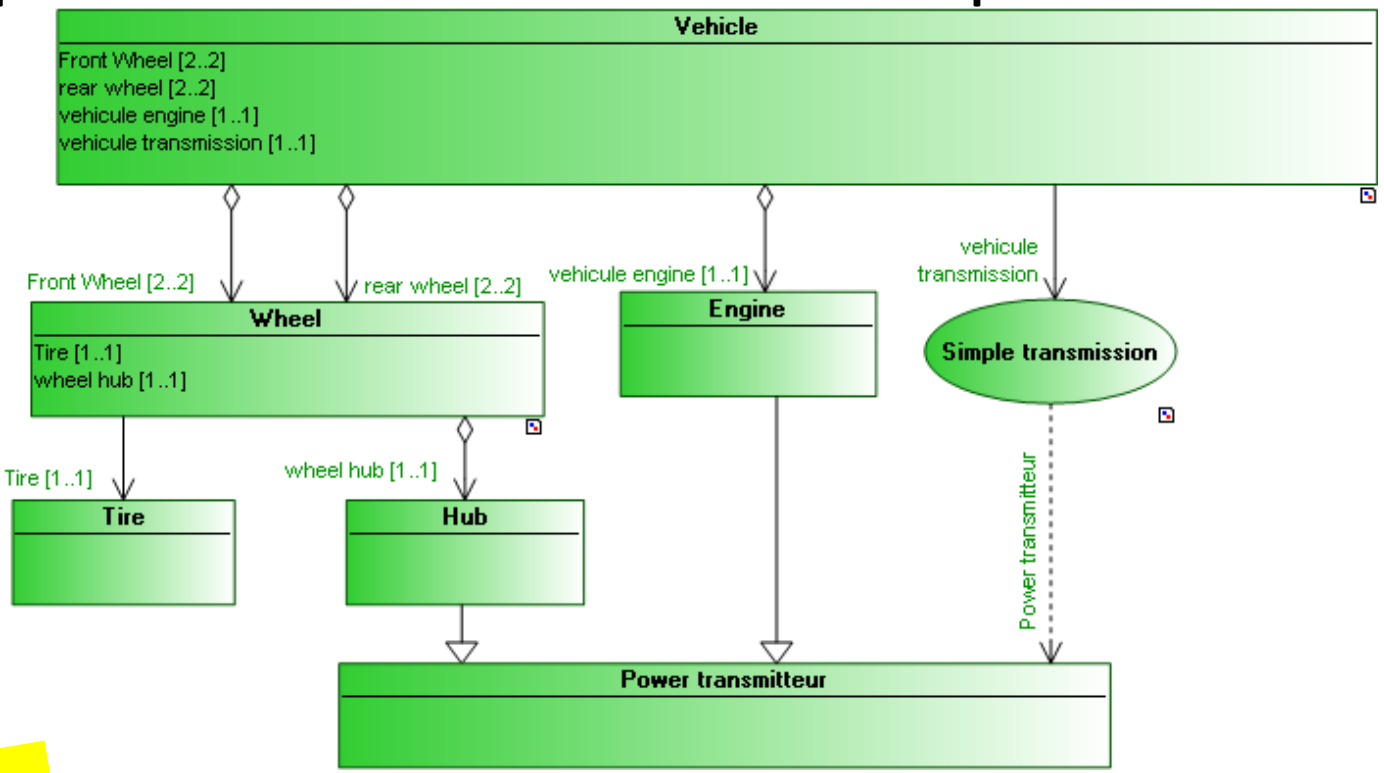
En logique formelle :

∀ s ∃ t (s (PERSONNE-6, CATÉGORIE-6) = t) (s ∈ S, t ∈ T.)

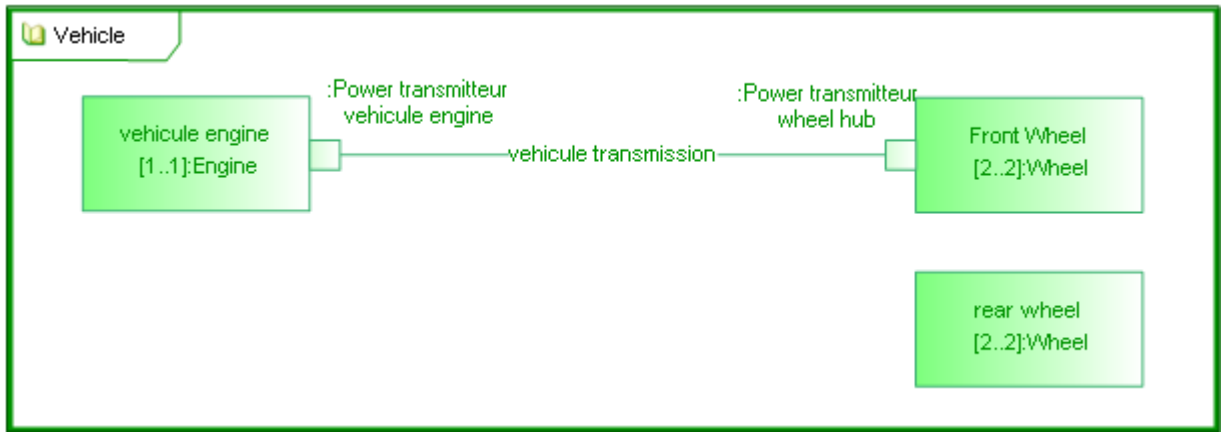


Au-delà de Merise
Yves Tabourier
1986

One step forward : Car example



Not available Yet
(Prototype in the conceptual MetaModel)

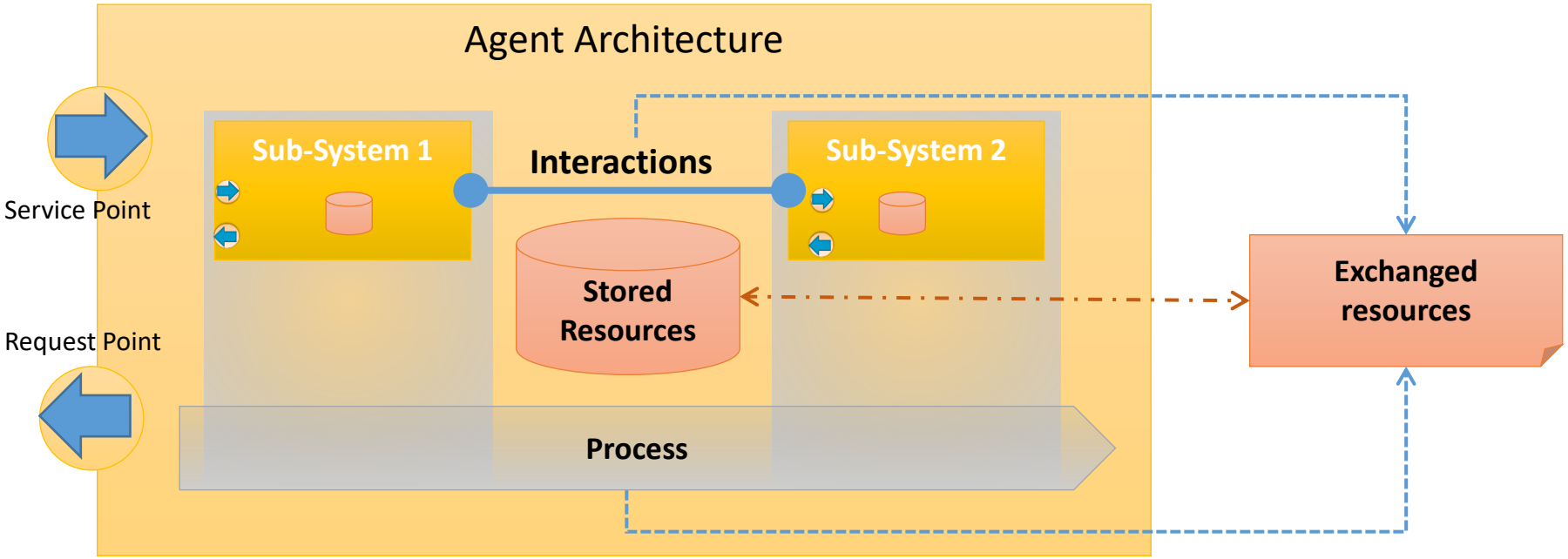


Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Roles
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

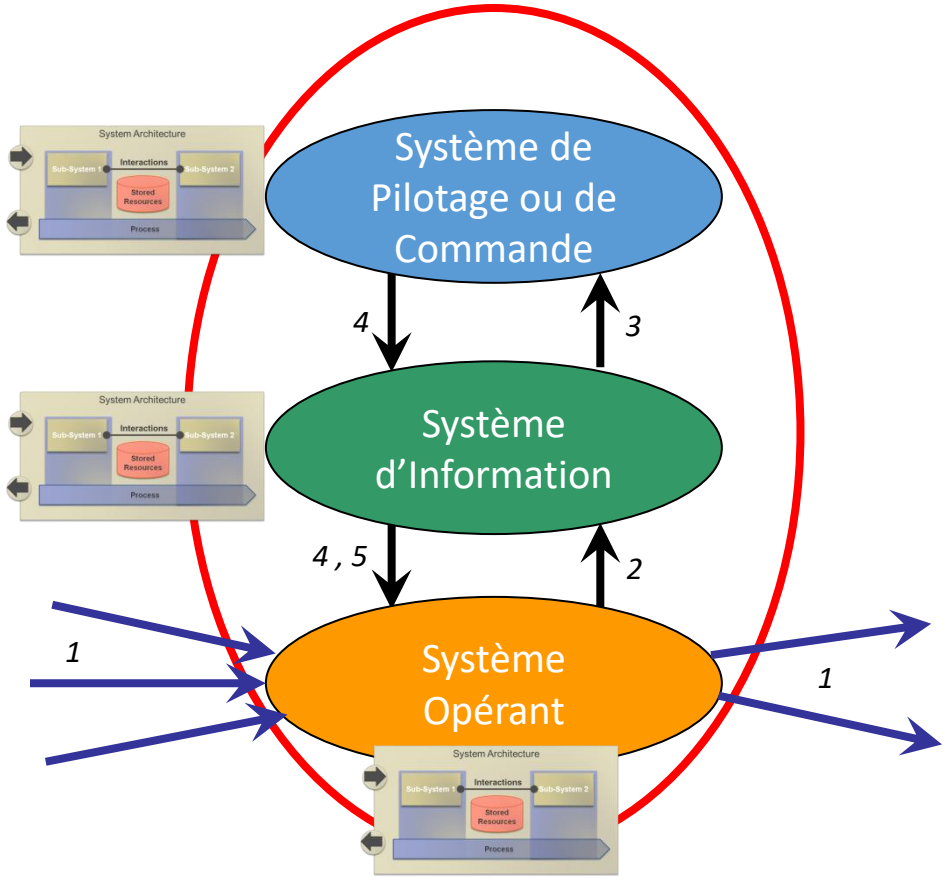
System architecture viewpoints

- Processing: actions, inputs, outputs.
- Communications: interactions, service points
- Results: transformed resources, exchanged resources



System

« Ensemble d'éléments structurés, en interaction dynamique et organisé en fonction d'un but »



1 : Flux échangés avec l'environnement (dont les flux d'information).

2 : Informations représentatives brutes.

3 : Évènements importants et informations synthétiques.

4 : Informations incitatives du pilote.

5 : Informations mémorisées ou calculées utiles au S.O.

Agenda

- Problem statement
- Relationship direction
- Object scope
- Object usage
- Sub typing & redefinition
- Roles
- Power types
- Structured relationships
- Information architecture & systemic architecture
- Benefits

Information mapping before modeling rules

- Source :
  - Matthew West, Developing High Quality Data Models

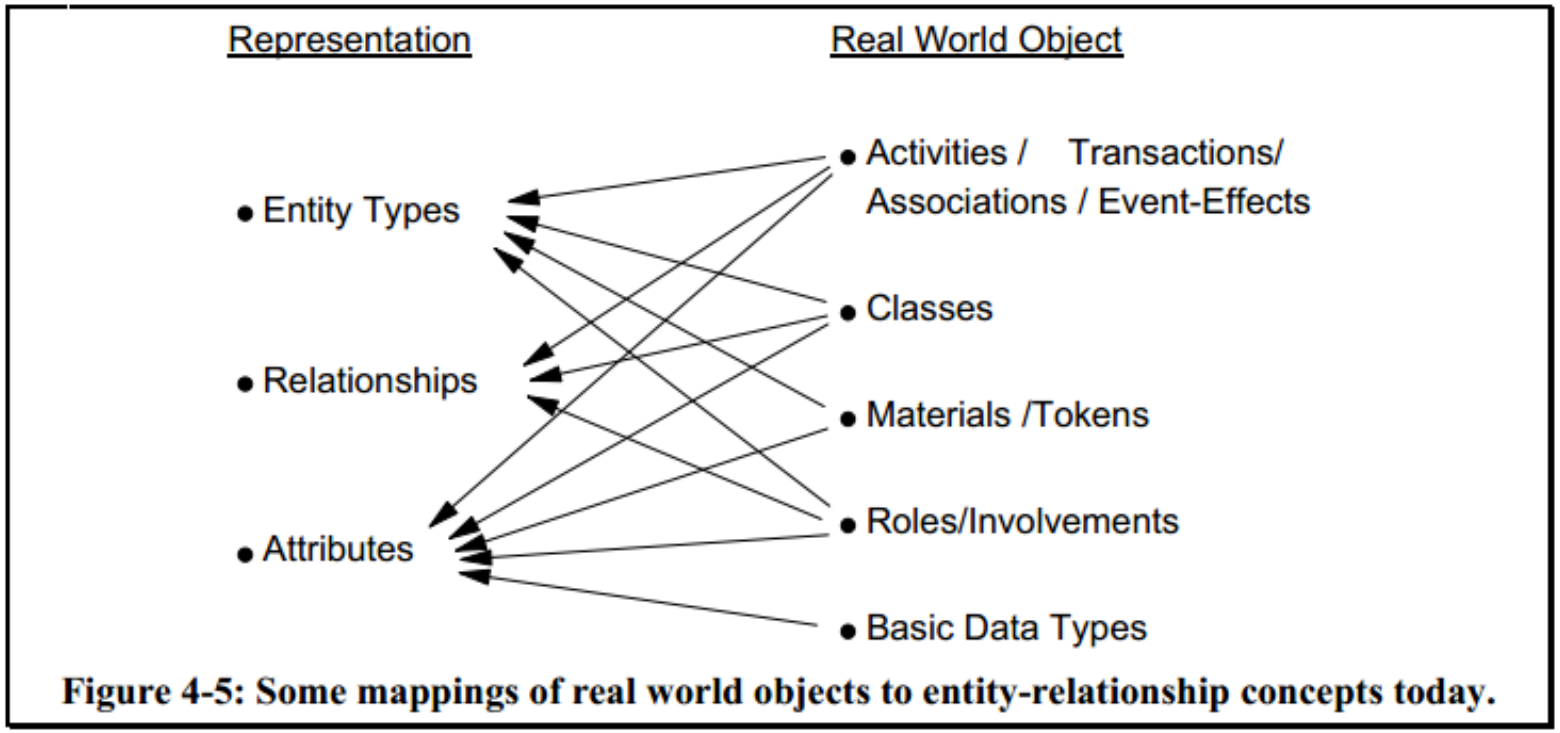


Figure 4-5: Some mappings of real world objects to entity-relationship concepts today.

Information mapping applying modeling rules

- Applying ISO 15926 rules

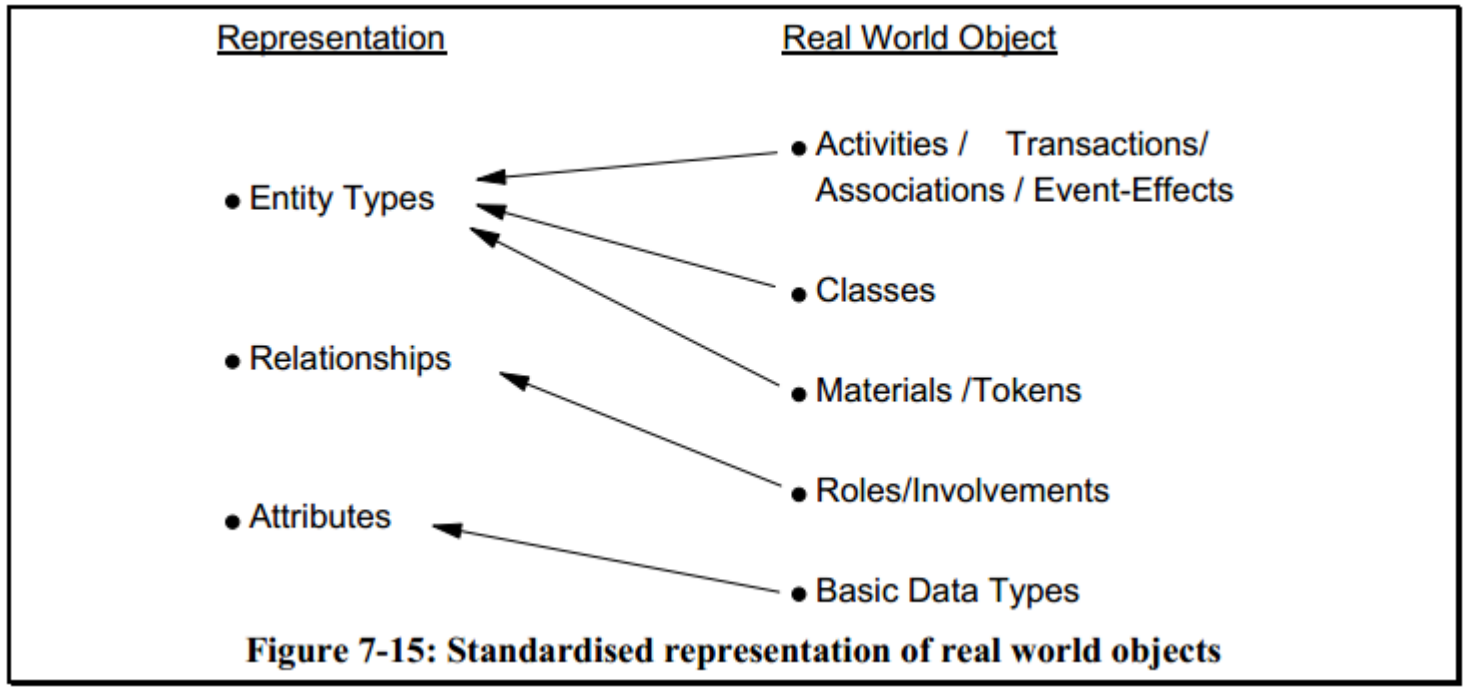


Figure 7-15: Standardised representation of real world objects

Benefits

- Well defined entities boundaries
  - Sound basis for establishing units of management (data domain boundaries, transaction boundaries, system boundaries, behavior boundaries, …)
  - Controlled approach for decomposition layers
    - Whole / Parts -> Whole / Parts -> …..
  - Controlled side-effects when changing a model element
  - Enhanced and controlled generalization mechanism (sub-types of whole / sub-types of parts).

- Issue of adopting a more abstract approach
  - Need to master more levels of abstraction
  - Abstraction is sometimes a difficult skill

- Benefits of adopting a more abstract but consistent approach
  - Incremental adoption when coming from traditional E/R approaches
  - Dramatic reduction of core principles
    - Whole / Parts
    - Super-type / sub-types
    - Redefinition
    - Part Inter connections
  - The unification of core constructs counter-balances the issue of additional abstraction

APPENDIXES

Data Accurary

Dimensions of Availability

Does the system do what it is supposed to do?

Functionality

Is the data provided by the system accurate and complete?

Data Accuracy

Performance

Does the system function within the acceptable performance criteria?

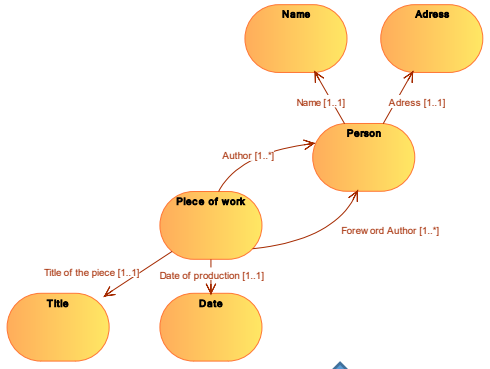
	COMMUNICAT.	DONNEES	TRAITEMENT
CONCEPTUEL	*M C C*	*M C D*	M C T
ORGANISATIONNEL	M O C	M O D	*M O T*
LOGIQUE	M L C	M L D	M L T
PHYSIQUE	M P C	M P D	M P T

Les classes et Data Domain de IA correspondent au MOD/MLD de Merise

Information Architecture Governance

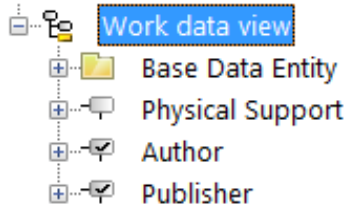
Business Architect

Concpetual Reference Model

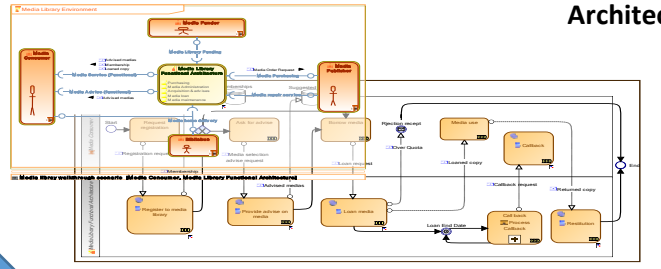


Data Scientist

Business Information Views



Business Information Flows

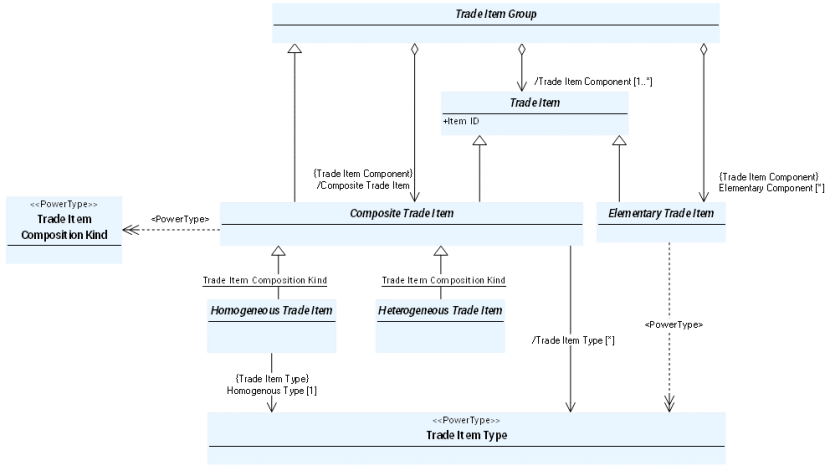


Business Architect

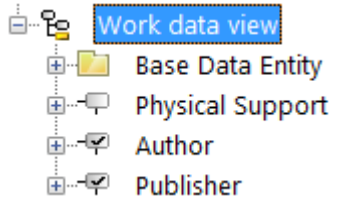
Business Architect

Information Architect

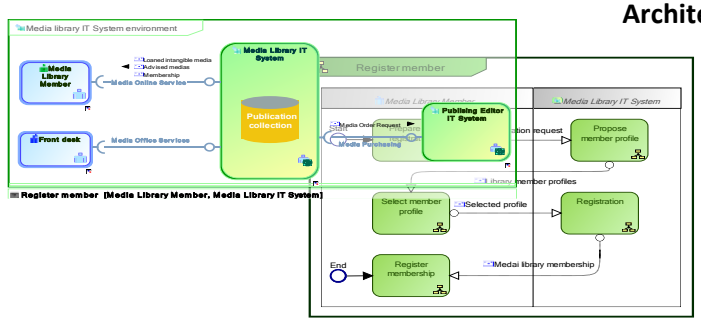
Application Data Reference Model



Data Views



Application Data flows



Application Architect

Data Architect

Data Scientist

- ...what the analog gains in semantics it loses in syntactics, and what the digital gains in syntactics it loses in semantics. Thus, it is that because the analog does not posses the syntax necessary to say "No" or to say something involving "not" one can refuse or reject in the analog, but one cannot deny or negate' (Wilden 1980: 163).

- The trick of digitalization consists in introducing gaps into continuums, thereby creating boundaries. These boundaries, however, do not themselves belong to the continuum, neither are they part of the gap. The boundary is the locus of an external intervention and thus necessarily defines a goalseeking system that drew that boundary. Therefore a system of a higher logical type - defining the goal - is necessarily established in the process of digitalization.

- Finally, and most important perhaps, language (verbal or written) can speak about itself, it can meta-communicate (e.g. 'this sentence is in English' is a sentence about a sentence). Therefore messages in digital codes can belong to levels of different logical typing. For instance, animals may or may not be able to pretend, but 'only humans pretend to pretend' (Wilden 1980: 173).

- In order for this to be fruitful in normal life-situations the insights gained by these mental manipulations must be translatable back to the analog code of active life, that is change in the 'real' world. This in fact is the way of evolution in the cultural sphere: an unending *chain of coding* and recoding, translation, between digital and analog form, language and 'reality'.
- Code Duality and the semiotic of Nature
- Code-Duality and the Semiotics of Nature (nbi.dk)

Mais le modèle de Nijssen prévoit aussi de montrer des liens de nature ensembliste *entre les types* d'objets non lexicaux eux-mêmes, via la notion de sous-type. Cette possibilité ne se présente pas sur la figure 1d, c'est pourquoi je vais m'appuyer sur deux autres exemples (fig. 4).

En 4a, qui pourrait être vue comme une extension de 1d, on montre que le NOLOT « CLIENT » est *une partie* du NOLOT plus vaste « TIERS », qui inclut également le NOLOT « FOURNISSEUR ». En 4b, on voit de la même façon que le NOLOT « PERSONNE » contient à la fois « PERSONNE PHYSIQUE » et « PERSONNE MORALE », lesquels types d'objets sont *exclusifs* (ce qui est symbolisé par X), et forment la *totalité* des personnes (ce qui est symbolisé par T).

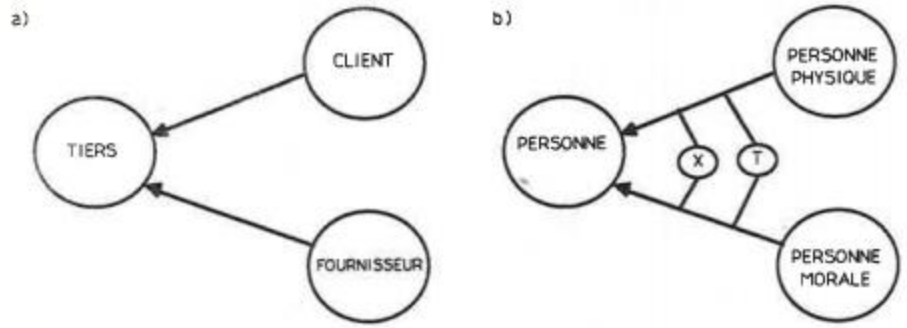
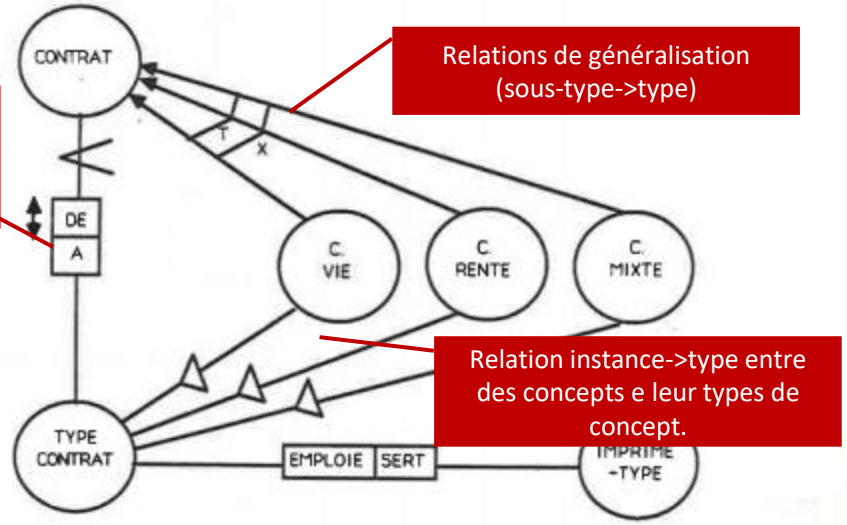


Figure 4. — *Liens de sous-types dans IA*

Dans ce survol je n'ai pas tout dit du formalisme IA, pas plus que des autres : assez cependant, je crois, pour pouvoir passer à la comparaison critique des quatre formalismes présentés.

13.4.1. IA (Nijssen)

IA illustre avec éclat l'attention portée au lien entre réel et données (C_3), sous la forme des ponts entre LOTs et NOLOTs. L'usage de cercles pleins entourés de pointillés n'est qu'un raccourci pour abréger les ponts privilégiés, mais ne saurait être interprété comme une ambiguïté sur le statut des objets. L'attention à la représentation du réel y est très forte aussi (C_2) : en ne gardant que les NOLOTs et les idées, on a sous les yeux une représentation de la structure des classes d'objets du monde réel. Non seulement les classes de rencontre entre objets particuliers y sont montrées, mais également des liens de nature ensembliste entre les catégories d'objets : inclusion, réunion (inclusion avec totalité sur les parties), partition (réunion avec exclusion) offrent une panoplie intéressante, qui pourrait être assez naturellement étendue. Des liens entre les idées, de même nature, y existent aussi.



Ensemble de toutes less catégories de contrat

Relations de généralisation (sous-type->type)

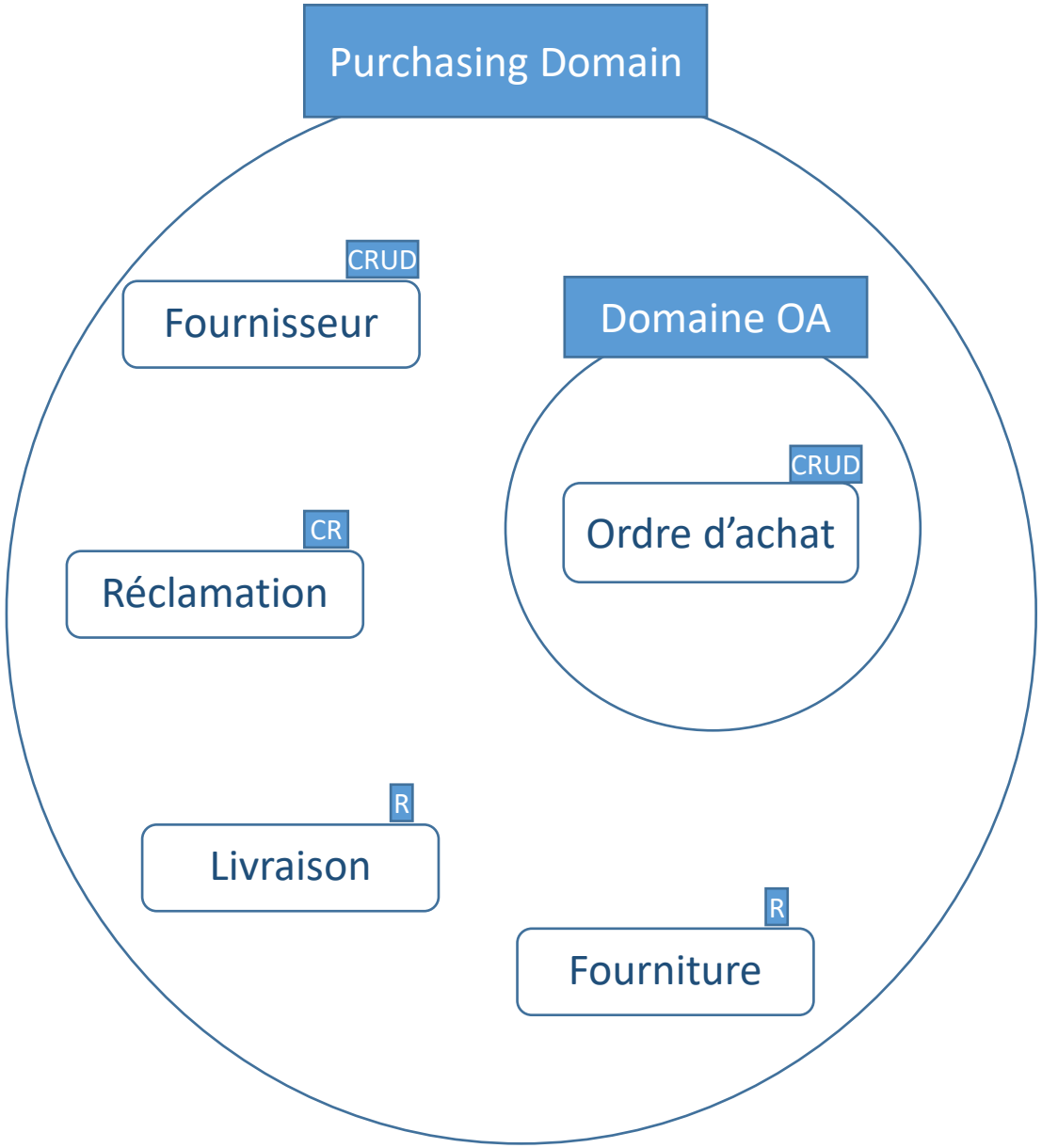
Relation instance->type entre des concepts e leur types de concept.

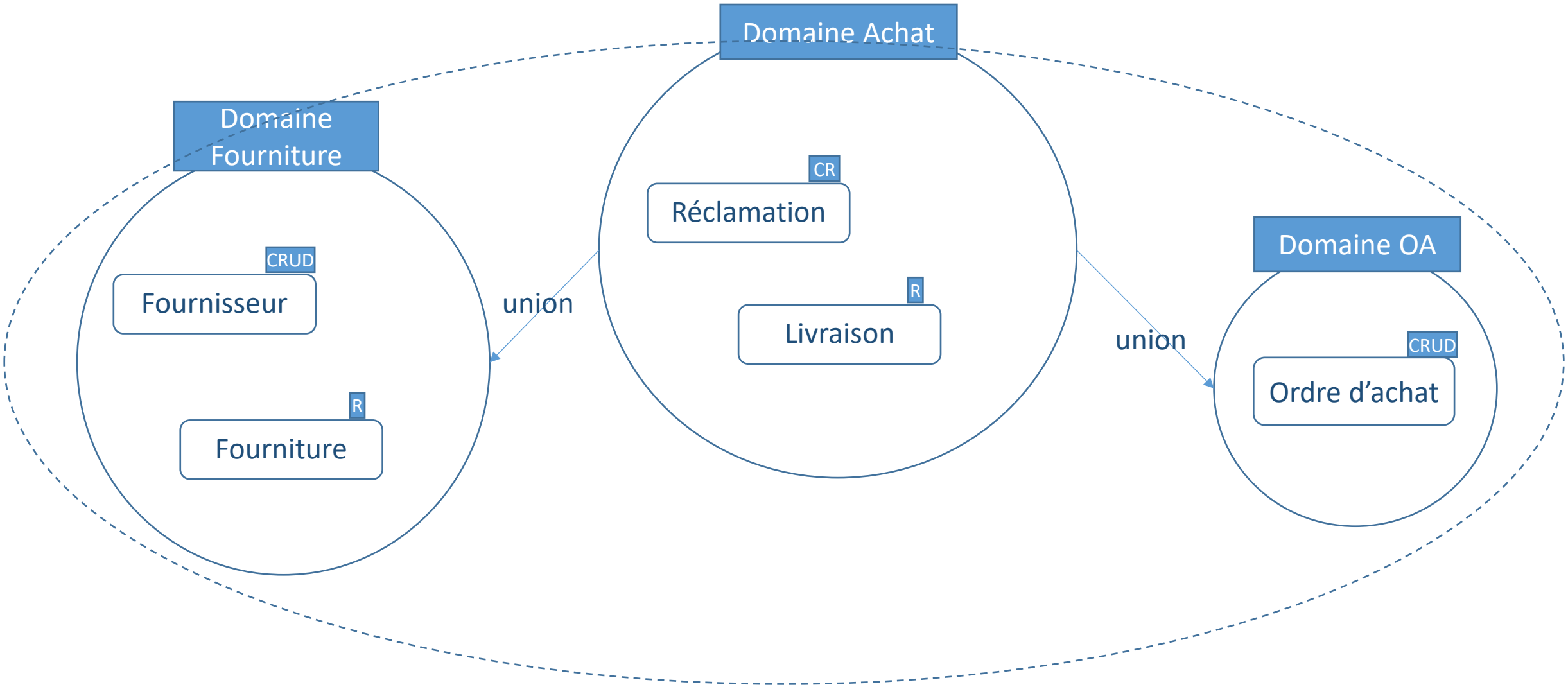
Figure 5. — *Des types d'objets, occurrences d'un autre type. A noter que l'idée DE A devient pratiquement inutile*

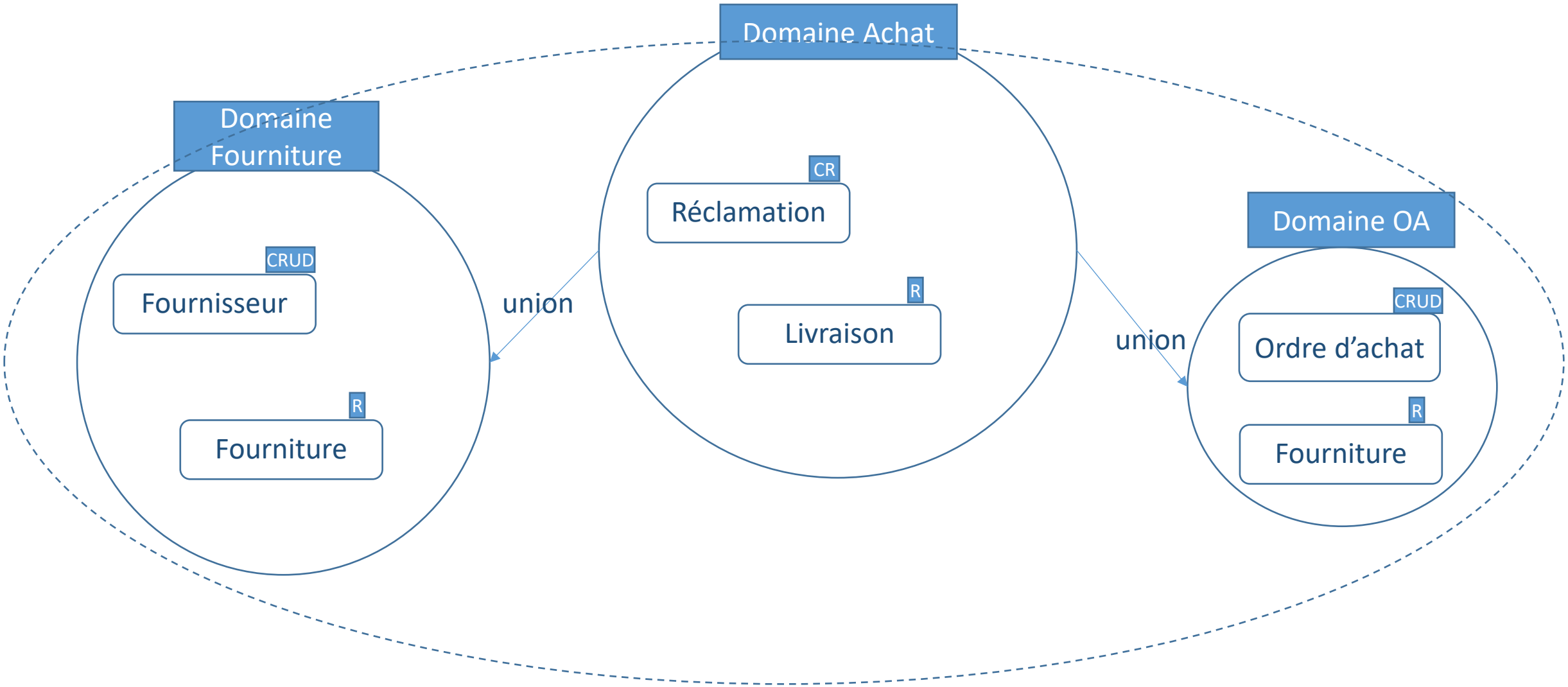
Une extension intéressante de cette orientation pourrait même être introduite : être capable de dire qu'une catégorie d'objet *est un élément* d'une autre catégorie d'objet (fig. 5), ce qui revient à introduire la notion *d'ensemble d'ensembles* : je reviendrai sur ce thème plus loin

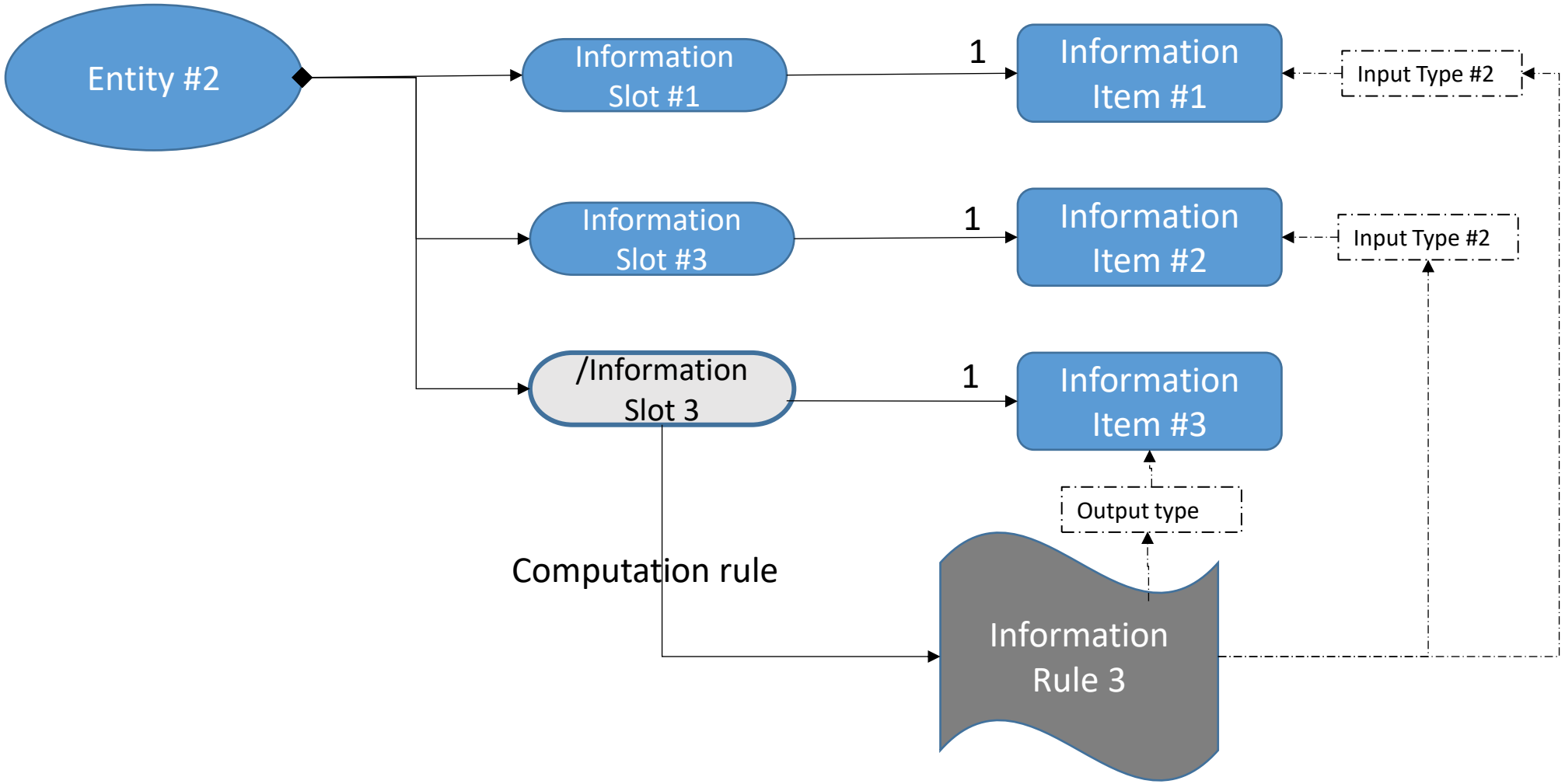
Nous faisons l'inverse. Nous allons calculer la relation instance-type avec Contrat-Type selon la règle suivante:

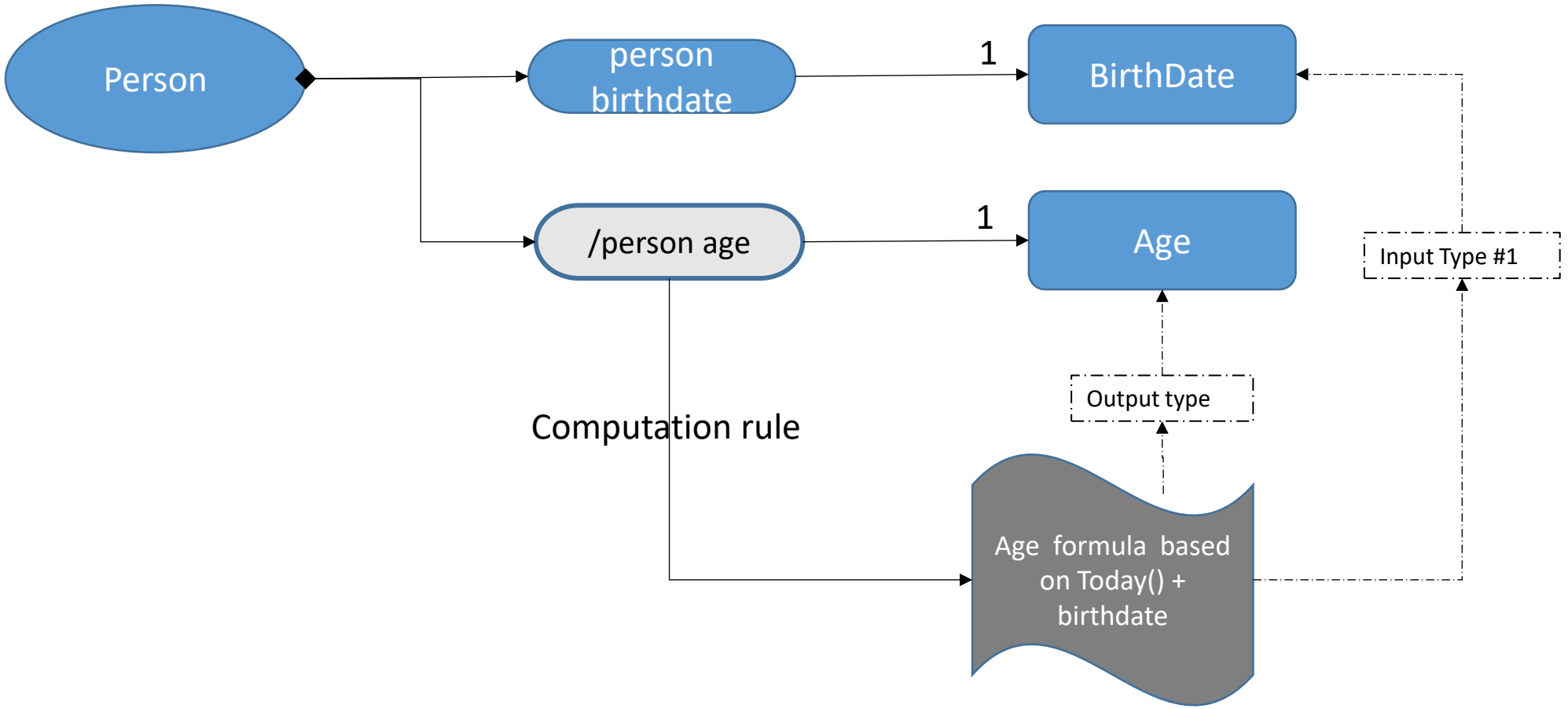
- Tous les sous-types d'un concept sont les instances de son concept type.

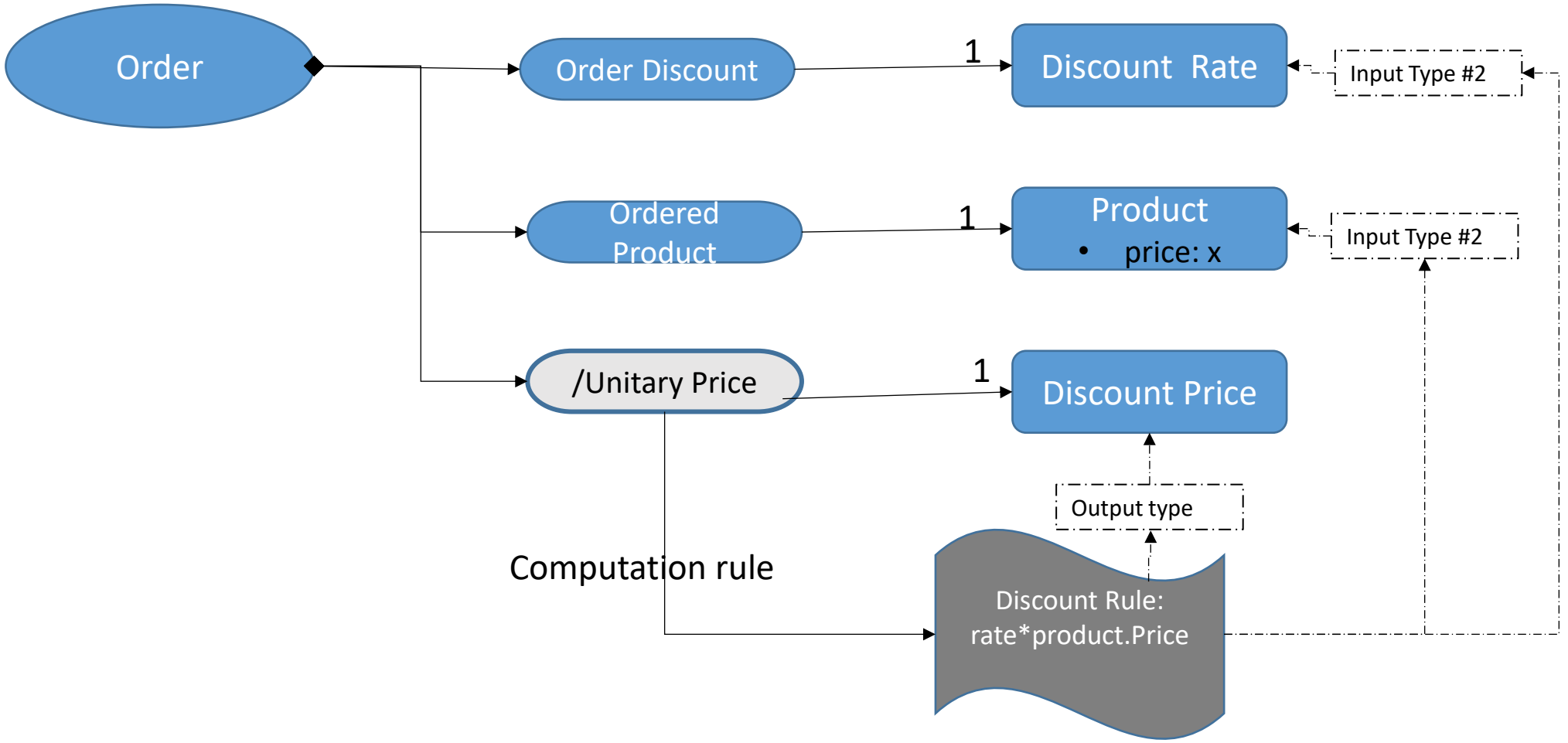

Purchasing Domain
CRUD
Fournisseur
Domaine OA
CRUD
Ordre d’achat
CR
Réclamation
R
Livraison
R
Fourniture


Domaine Achat
Domaine Fourniture
CR
Réclamation
CRUD
Fournisseur
union
R
Livraison
Domaine OA
union
CRUD
Ordre d'achat
R
Fourniture

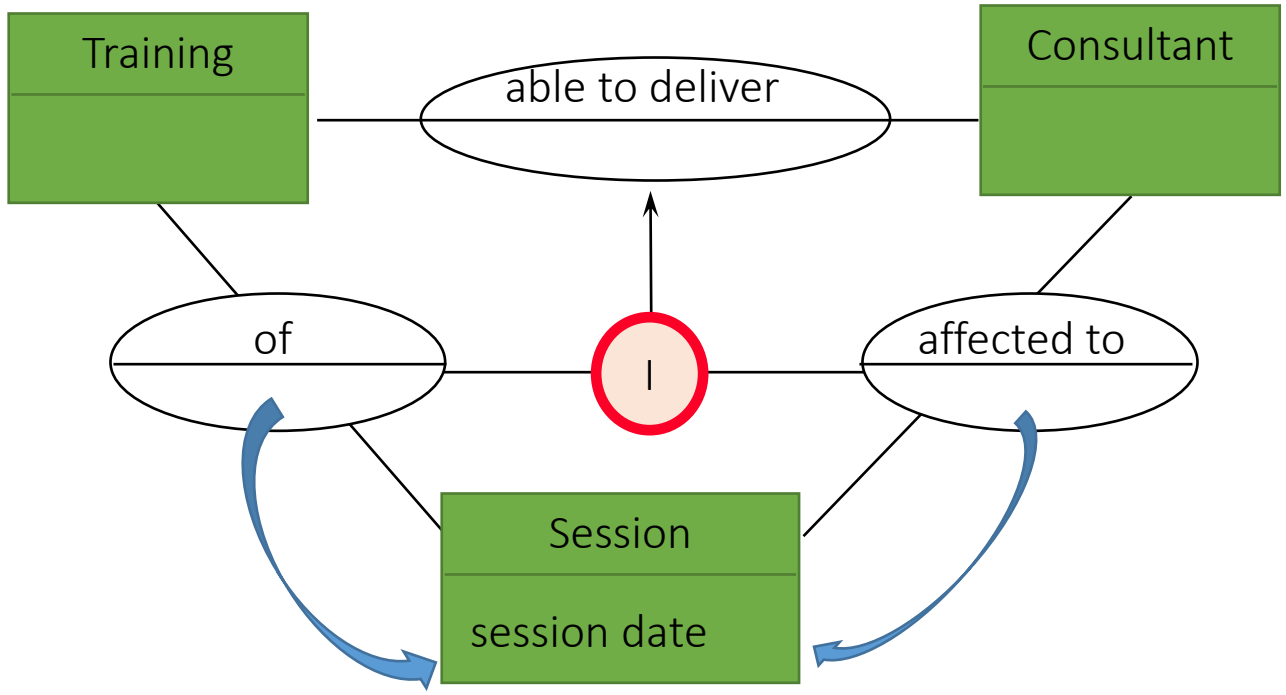

Domaine Achat
Domaine Fourniture
CR
Réclamation
CRUD
Fournisseur
R
Livraison
union
R
Fourniture
union
Domaine OA
CRUD
Ordre d’achat
R
Fourniture


Entity #2
Information Slot #1
Information Slot #3
/Information Slot 3
1
1
1
Information Item #1
Information Item #2
Information Item #3
Input Type #2
Input Type #2
Output type
Computation rule
Information Rule 3


Person
person birthdate
1
BirthDate
/person age
1
Age
Input Type #1
Output type
Computation rule
Age formula based on Today() + birthdate

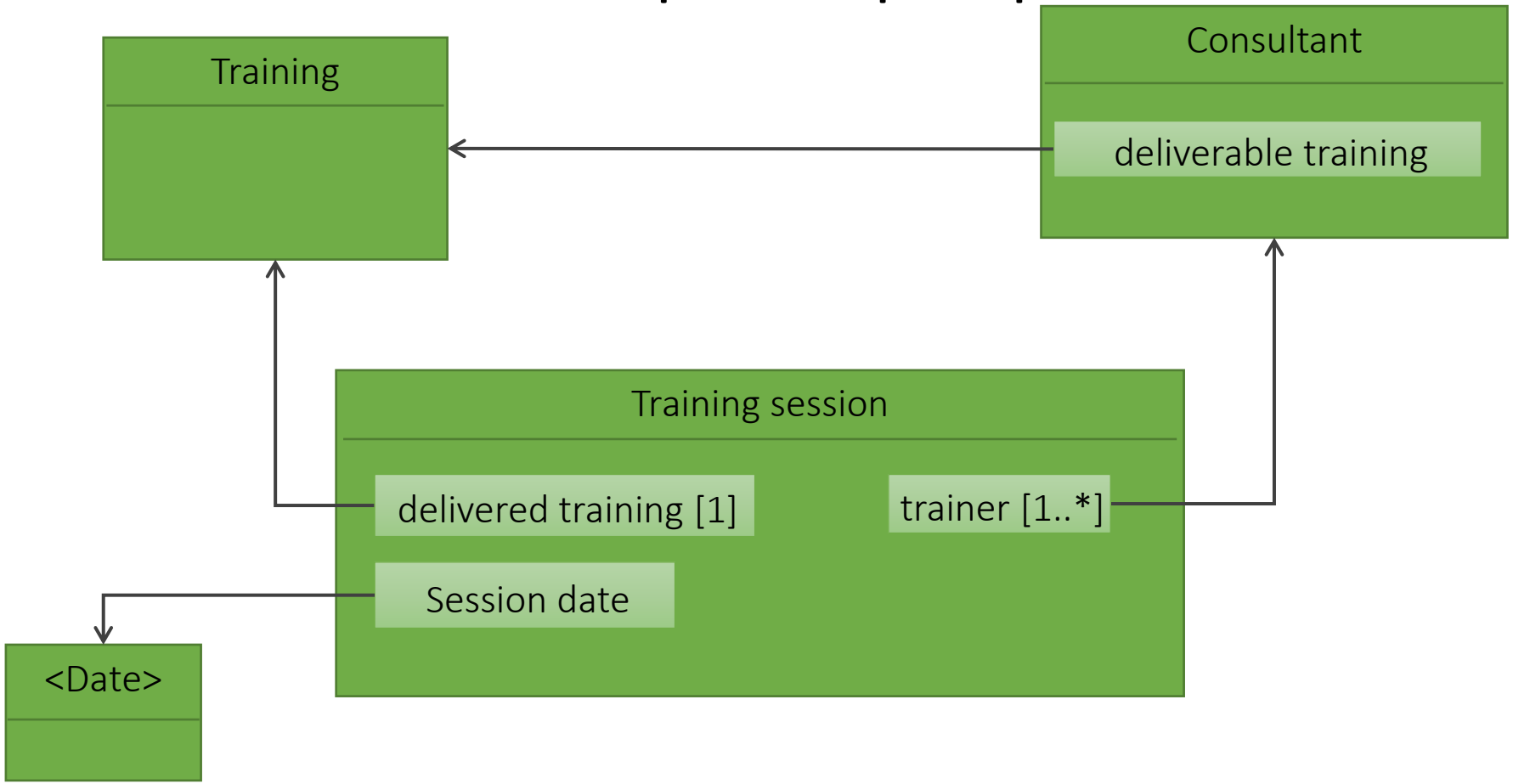

Order
Order Discount
1
Discount Rate
Input Type #2
Ordered Product
1
Product
• price: x
Input Type #2
/Unitary Price
1
Discount Price
Output type
Computation rule
Discount Rule:
rate*product.Price

Entities & Relationships



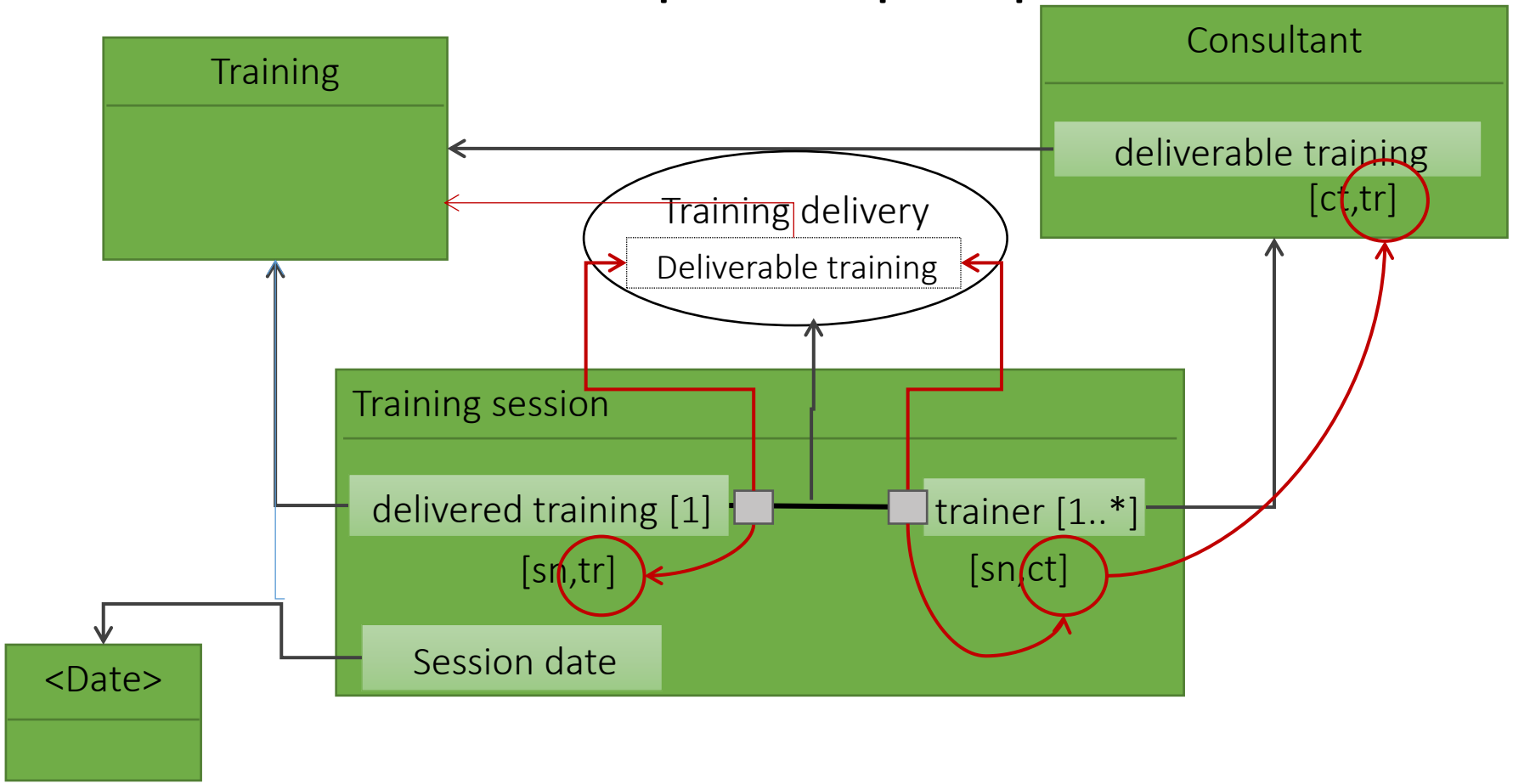
- Consultants can provide trainings
- Trainings are delivered offering sessions, at a particular date, by consultants

From relationships to properties



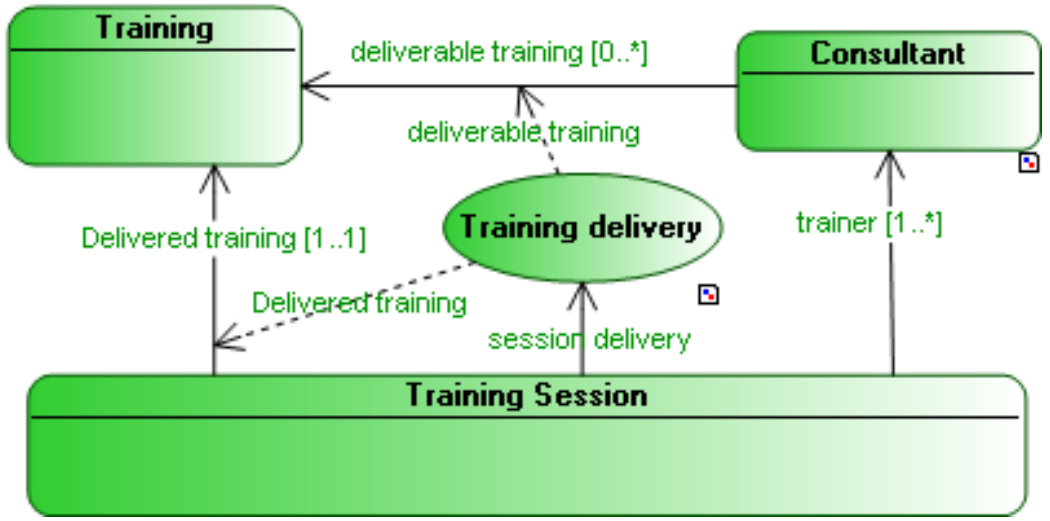
- Roles are played within composite structures

From relationships to properties

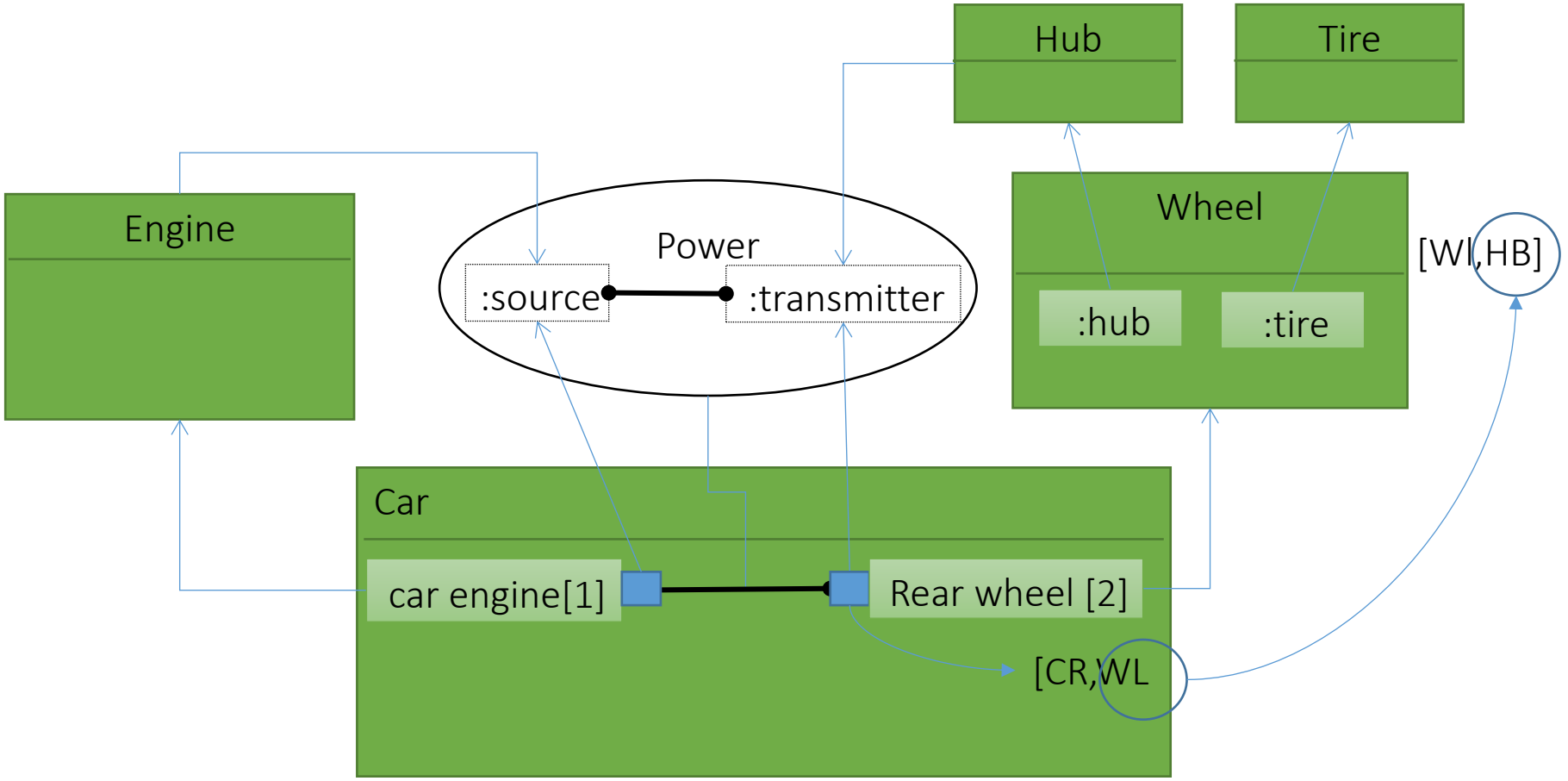


- Relationship types express constraints
- Relation types apply between roles in composite structures

Enhanced dictionary model in MEGA



One step forward : Car example



- The hub is the transmitter of the rear wheel

Directed Relationships & Structure & approaches

Directed Relationships & Structure	Replacement	Rational
Conceptual Entity	Information Concept	The Business Information Layer has now its own meta-model handled by IA Concepts.
Data Model	Information Domains Information Views Logical Data Domains Application Data Domains	Data Model was mixing three different semantics: 1. Structure of information entities 2. View of information entities 3. Scope of entities allocated to systems.
Entity (DM)	Class	The UML Class Model has been adopted by the majority of users Our current implementation of the (DM) model cannot benefit from the advanced “part model” of UML already available in MEGA.
Association (DM)	UML Part	In the E/R model, Associations and Entities are both nodes in the Data Graph. Hence, Association (relationships) are considered as “external” to entities, preventing from defining a proper “scope” for Entities.
Systems -> Data Models	System -> Information Store	This use case for Data Models has been replaced by Information Stores.
System -> Entity CRUD	Information Stores-> Information Entity CRUD	“Information Domain” provide a grouping a related Information Entities with CRUD characteristics. CRUD is not more managed Entity by Entity, for each considered System.

Operating Model, Capabilities and Information

- **Information is defined:** it has structure and meaning (semantic definition).
- **Information is exchanged** – between interacting agents.
- **Information is processed** – though processes occurring under agent responsibilities.
- **Information is memorized** – in stores under agent responsibilities.

- **Capabilities** define information involved in offered services.