ArchiMate - The Open Group Standard

Barbara Re

What is ArchiMate?

- ArchiMate is a **modelling technique** ("language") for describing enterprise architectures

- It presents a clear set of **concepts** within and **relationships** between architecture domains, and offers a simple and uniform structure for describing the contents of these domains

- ArchiMate offers a **common language** for describing the construction and operation of **business** processes, **organizational structures**, **information flows**, **IT systems**, and **technical infrastructure**

- This insight helps the different stakeholders to design, assess, and communicate the consequences of **decisions** and **changes** within and between these business domains

What ArchiMate provides

- A **language** with concepts to describe architectures
- A **framework** to organize these concepts
- A **graphical** notation for these concepts
- A **vision** on visualizations for different stakeholders
- An **open standard** maintained by the Open Group

- **Why a new version of ArchiMate (i.e. why ArchiMate 3.0)?**
- Increasing demand for relating EA to business strategy
- Technology applications that mix IT and physical world
- Usage in new domains, e.g. manufacturing, logistics
- Improved consistency and comprehensibility
- Improved alignment between ArchiMate and TOGAF

ArchiMate® 3.0.1 Specification, *an Open Group Standard*

Description and Example are taken from

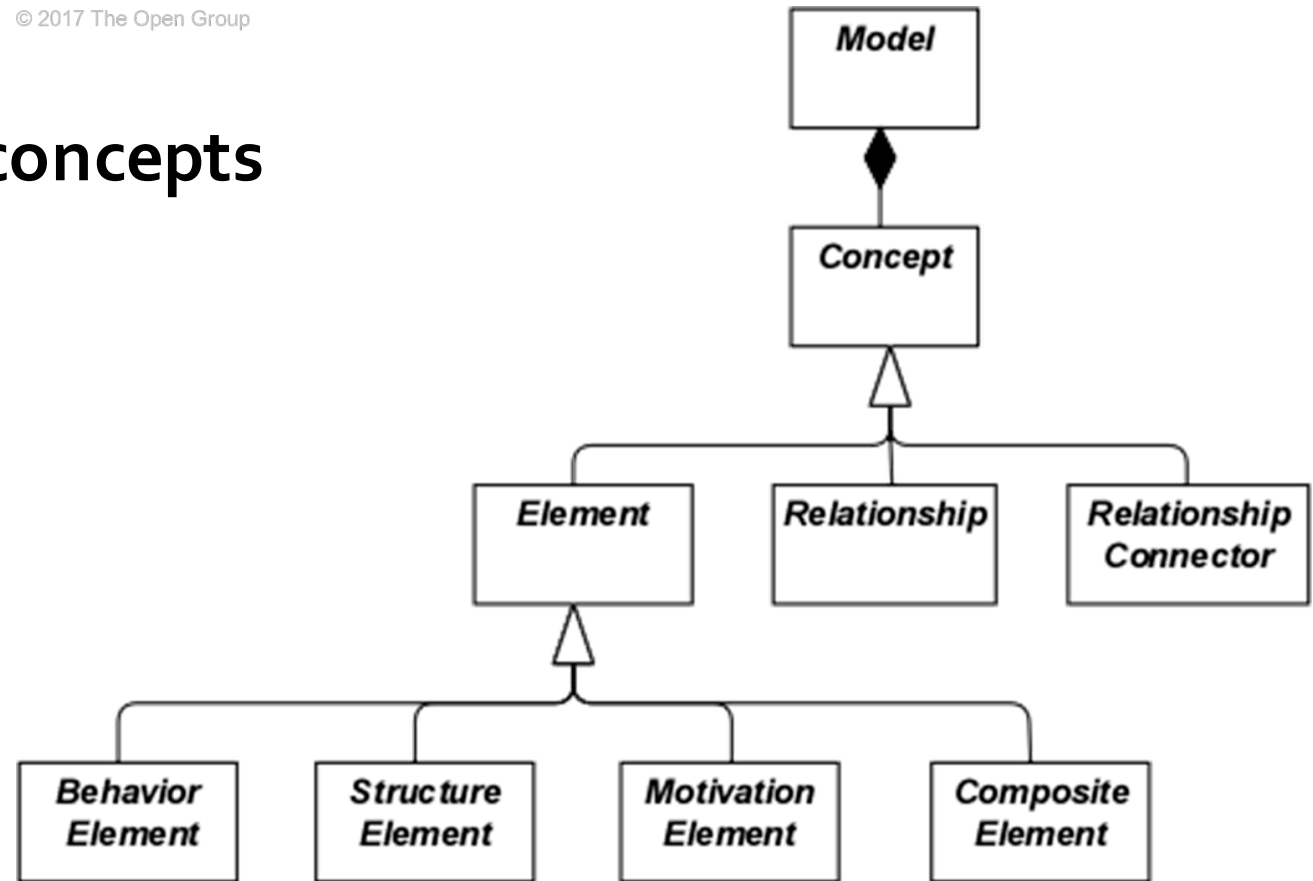http://pubs.opengroup.org/architecture/archimate3-doc/toc.html
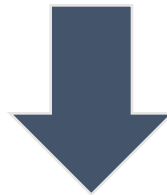
Top-level hierarchical structure of the language

© 2017 The Open Group

- A model is a collection of **concepts**
- A concept is either
  - an **element**
    - a behavior element
    - a structure element
    - a motivation element
   - a composite element
  - a **relationship**

Archimate 3.0.1 Specification, page 6

Three layers

- The *Business Layer* depicts **business services offered to customers**, which are realized in the organization by business processes performed by business actors

- The *Application Layer* depicts **application services that support the business**, and the applications that **realize** them

- The *Technology Layer* depicts **technology services** such as **processing, storage, and communication services needed to run the applications**, and the **computer** and **communication** hardware and system software that realize those services

© 2017 The Open Group
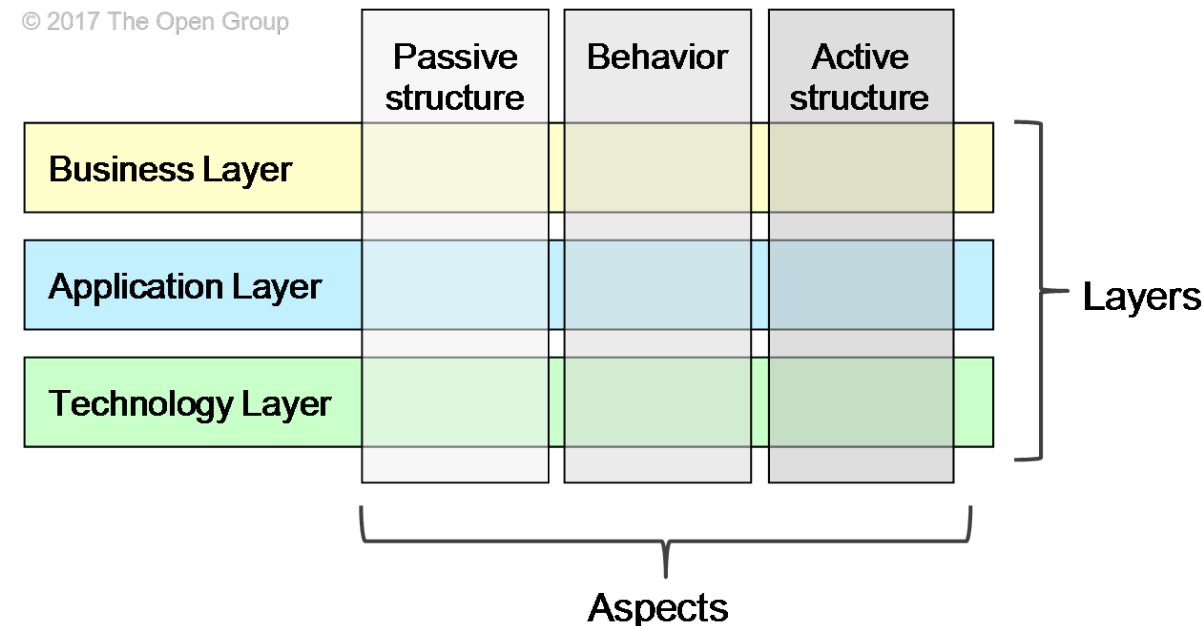
Three aspects

- The *Active Structure Aspect*, which represents the **structural elements** (the business actors, application components, and devices that display actual behavior; i.e., the "subjects" of activity)

- The *Behavior Aspect*, which represents the **behavior** (processes, functions, events, and services) performed by the actors. _Structural elements are assigned to behavioral elements, to show who or what displays the behavior_

- The *Passive Structure Aspect*, which represents the **objects on which behavior is performed**. _These are usually information objects in the Business Layer and data objects in the Application Layer, but they may also be used to represent physical objects_

© 2017 The Open Group

	Passive structure	Behavior	Active structure	
Business Layer				Layers
Application Layer				
Technology Layer				
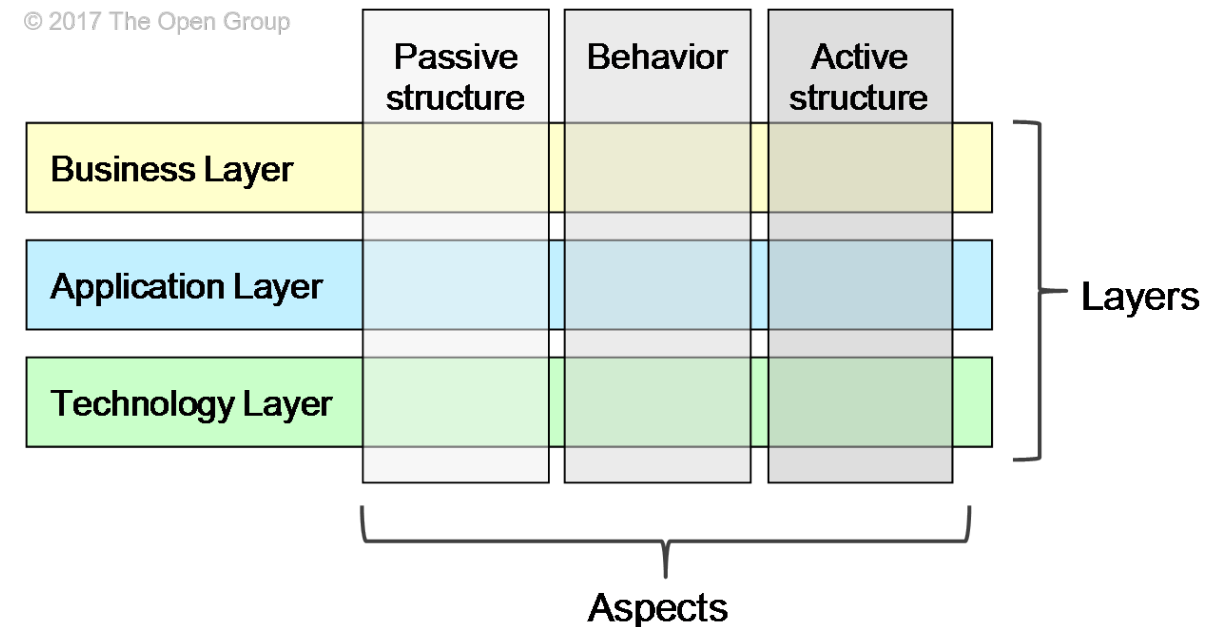
Aspects

Core Aspects in ArchiMate 3

■ Aspects correspond to a Subject-Verb-Object of sentences:

Full framework

	Passive structure	Behavior	Active structure	Motivation
Strategy				
Business				
Application				
Technology				
Physical				
Implementation & Migration				

Layers

Aspects
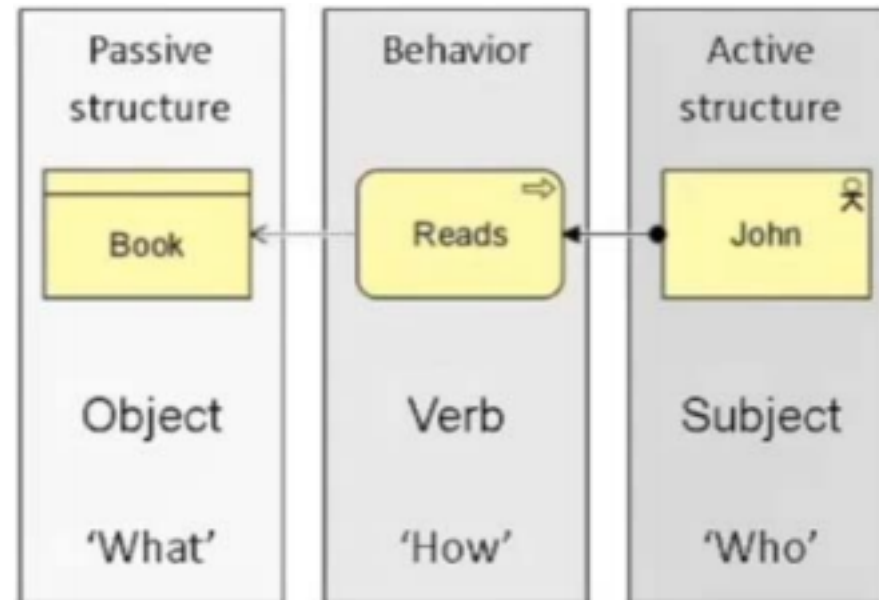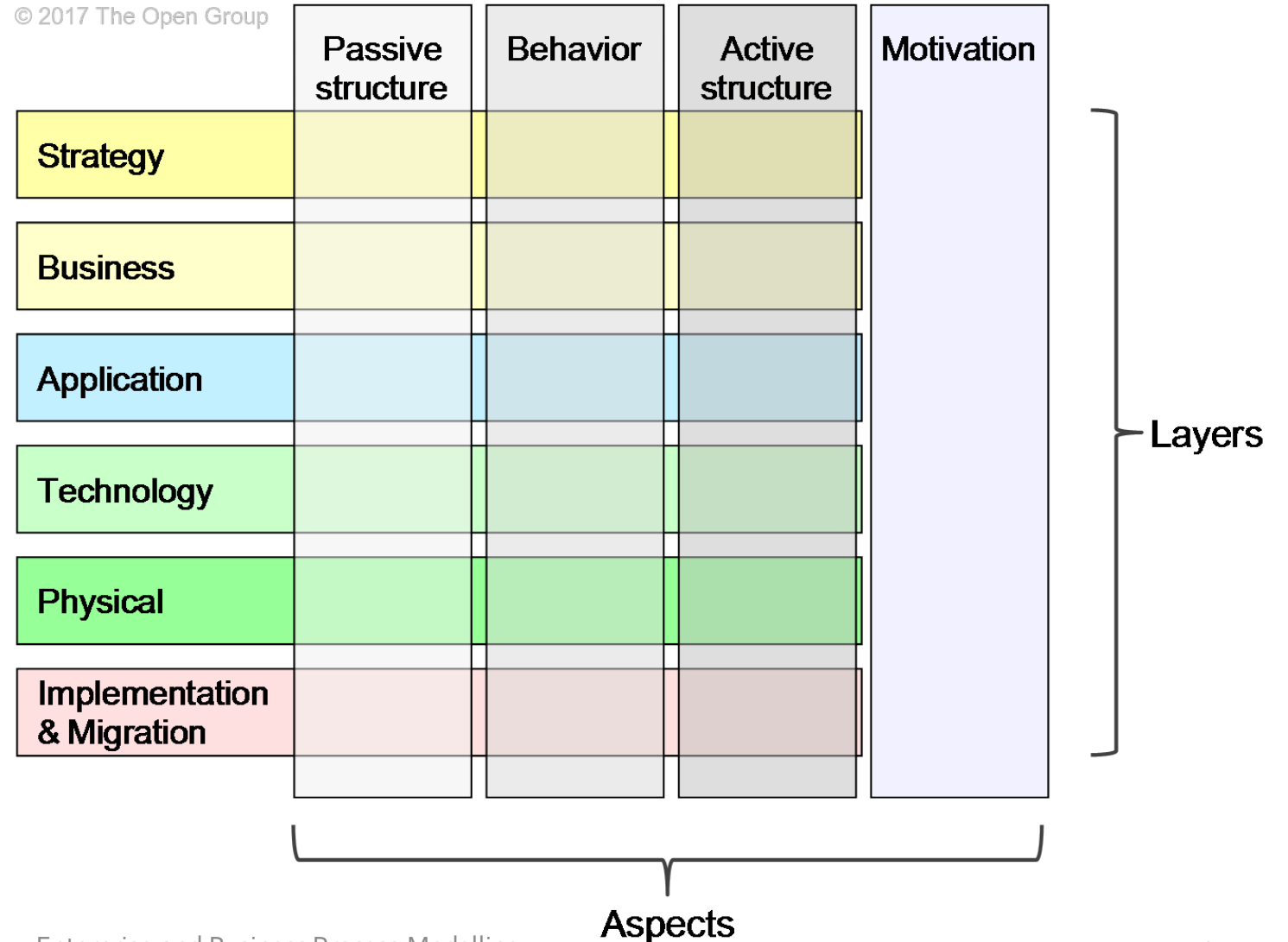
Full framework

- **Strategy Layer**
  - (*Capability, Resource, Course of Action*)
  - An ability that an active structure element, such as an organization, person, or system, possesses. An approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal. An asset owned or controlled by an individual or organization

- **Physical Layer**
  - (*Equipment, Facility, Distribution network, Material*)
  - An overview of the physical elements and their relationships, derived from the ArchiMate Technology layer

- **Implementation & Migration Layer**
  - (**Work package, Deliverable, Implementation event, Plateau, Gap**)
  - Focusing on the actual implementation of an EA and the migration process with work packages and dependencies.
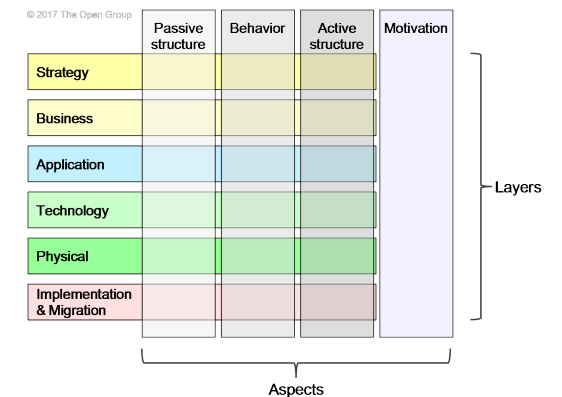
Generic metamodel

© 2017 The Open Group
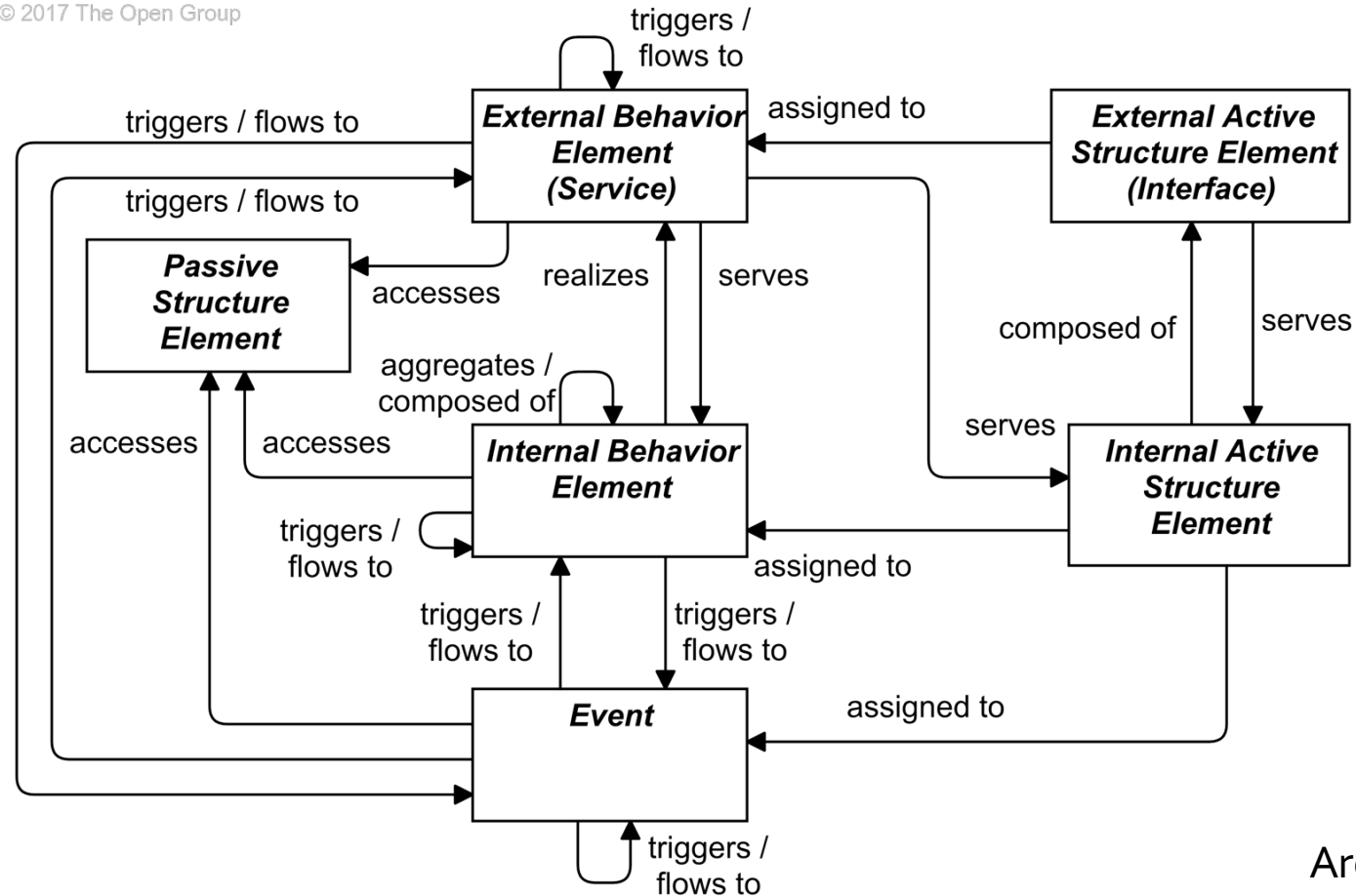
ArchiMate 3, section 4.1

Active structure elements

- An **internal active structure** element represents an entity that is capable of performing behavior

- An **external active structure element**, called an interface, represents a point of access where one or more services are provided to the environment
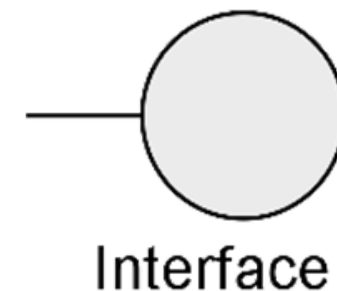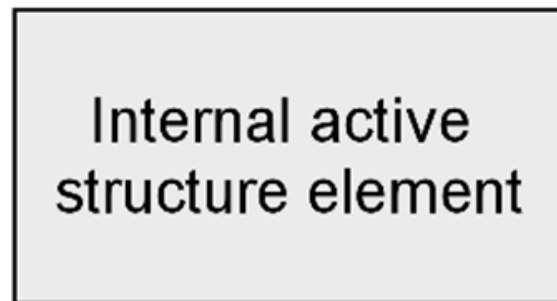
Behavior elements

- An internal behavior element represents a unit of activity performed by one or more active structure elements.
- An external behavior element, called a service, represents an explicitly defined exposed behavior.

- An event is a behavior element that denotes a state change.
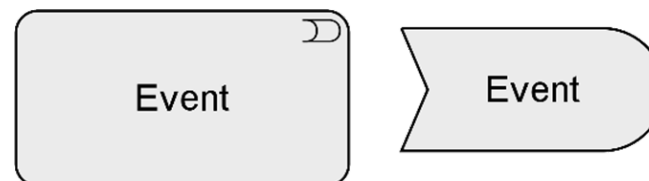
Passive structure elements

- Passive structure elements can be accessed by behavior elements

- A passive structure element is a structural element that cannot perform behavior. Active structure elements can perform behavior on passive structure elements

- Passive structure elements are often information or data objects, but they can also represent physical objects

Passive structure element
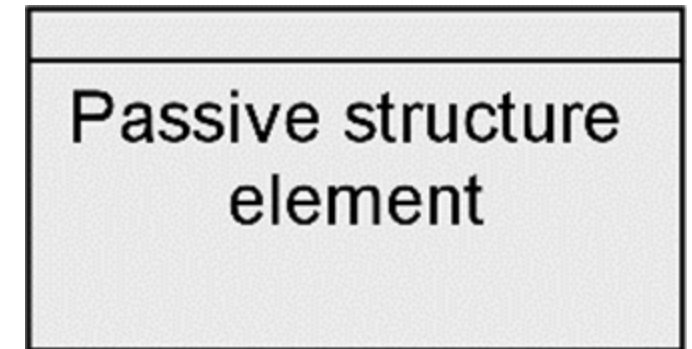
Generic collaboration and interaction notation

- A **collaboration** is an aggregate of two or more active structure elements, working together to perform some collective behavior

- An **interaction** is a unit of collective behavior performed by (a collaboration of) two or more active structure elements

Collaboration

Collaboration

Interaction

Interaction
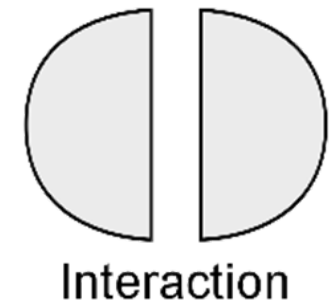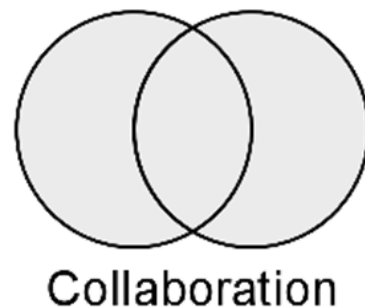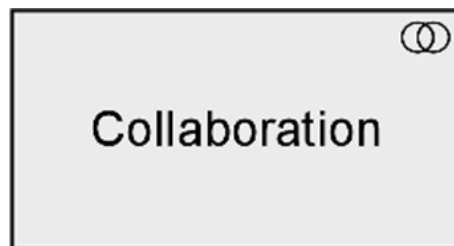
Generic process and function notation

- A process represents a sequence of behaviors that achieves a specific outcome

- A function represents a collection of behavior based on specific criteria, such as required resources, competences, or location

Motivation elements

Active Structure Element

Stakeholder

Motivation Element

Value

Meaning

Driver

Assessment

Goal

Outcome
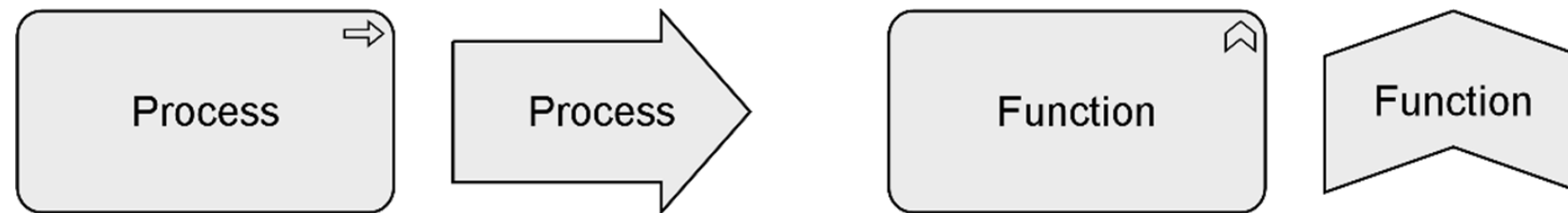
Principle

Requirement

Constraint

- Several *motivation elements* are included in the language: stakeholder, value, meaning, driver, assessment, goal, outcome, principle, and requirement, which in turn has constraint as a subtype

- A motivation element is an element that provides the context of or reason behind the architecture of an enterprise

Motivation element
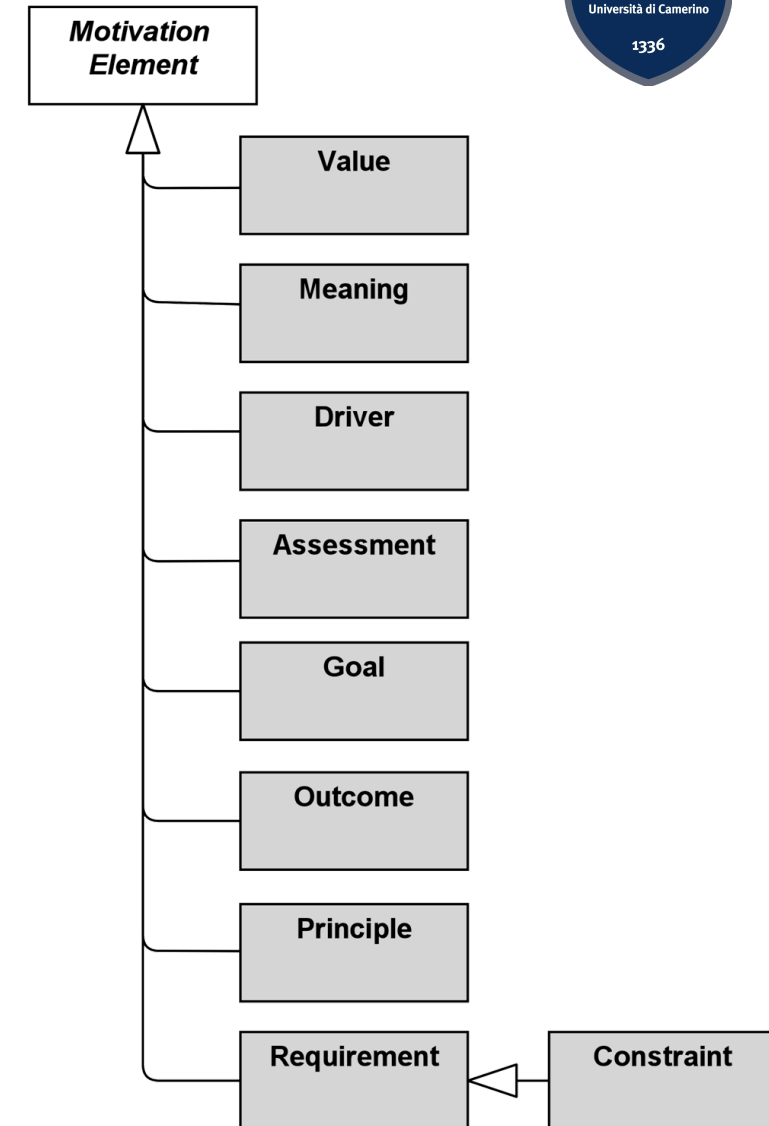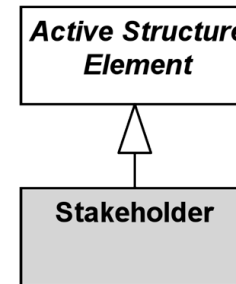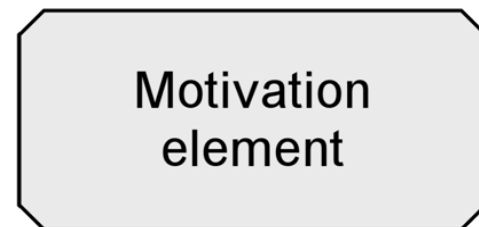
© 2017 The Open Group

Composite elements

- Composite elements consist of other concepts, possibly from multiple aspects or layers of the language.
- Grouping and location are generic composite elements
- Composite elements can themselves aggregate or compose other composite elements.

© 2017 The Open Group
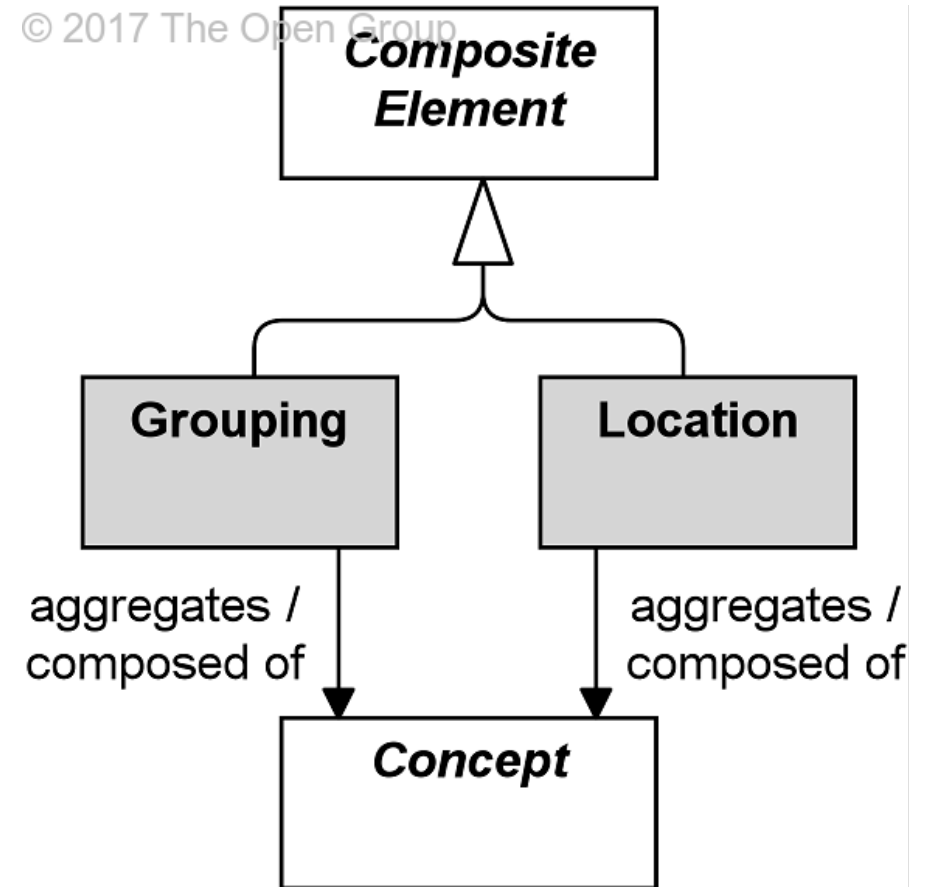
Grouping

- The grouping element aggregates or composes concepts that belong together based on some common characteristic.

In Example , the Grouping element is used to aggregate a conglomerate of two processes and an object that together realize a service (both with nesting and explicitly drawn aggregation relationships).

Enterprise and Business Process Modelling

19
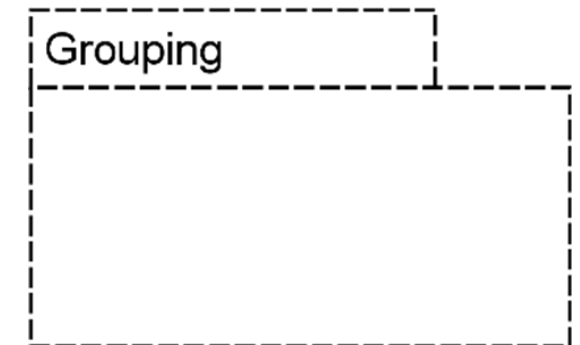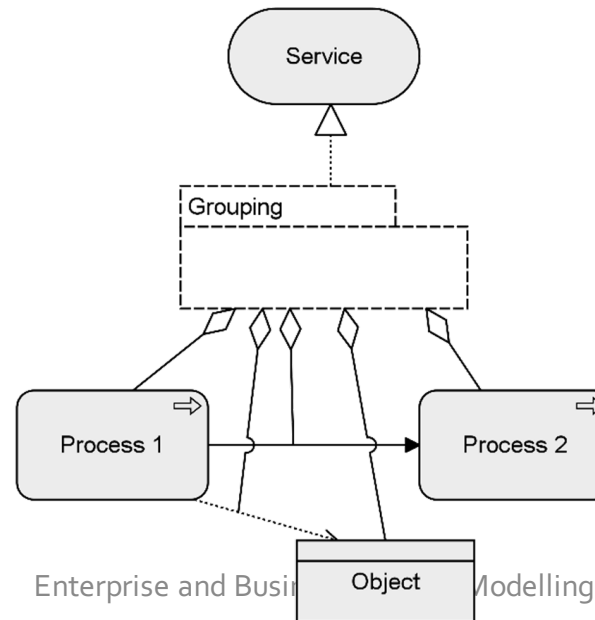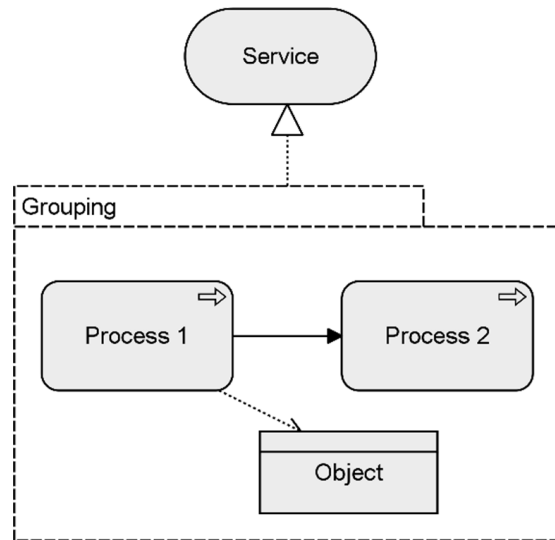
Location

- A location is a place or position where structure elements can be located or behavior can be performed.

- The location element is used to model the places where (active and passive) structure elements such as business actors, application components, and devices are located.

- This is modeled by means of an aggregation relationship from a location to structure element.

- A location can also aggregate a behavior element, to indicate where the behavior is performed.

- This element corresponds to the "Where" column of the Zachman framework

Location
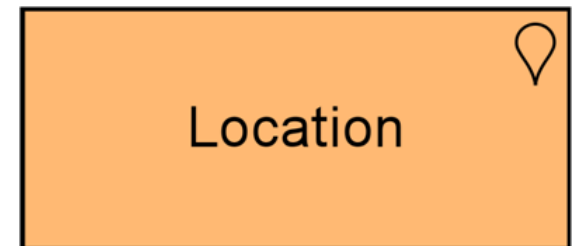
Relationships

- *Structural* relationships, which model the static construction or composition of concepts of the same or different types
- *Dependency* relationships, which model how elements are used to support other elements
- *Dynamic* relationships, which are used to model behavioral dependencies between elements
- *Other* relationships, which do not fall into one of the above categories

© 2017 The Open Group
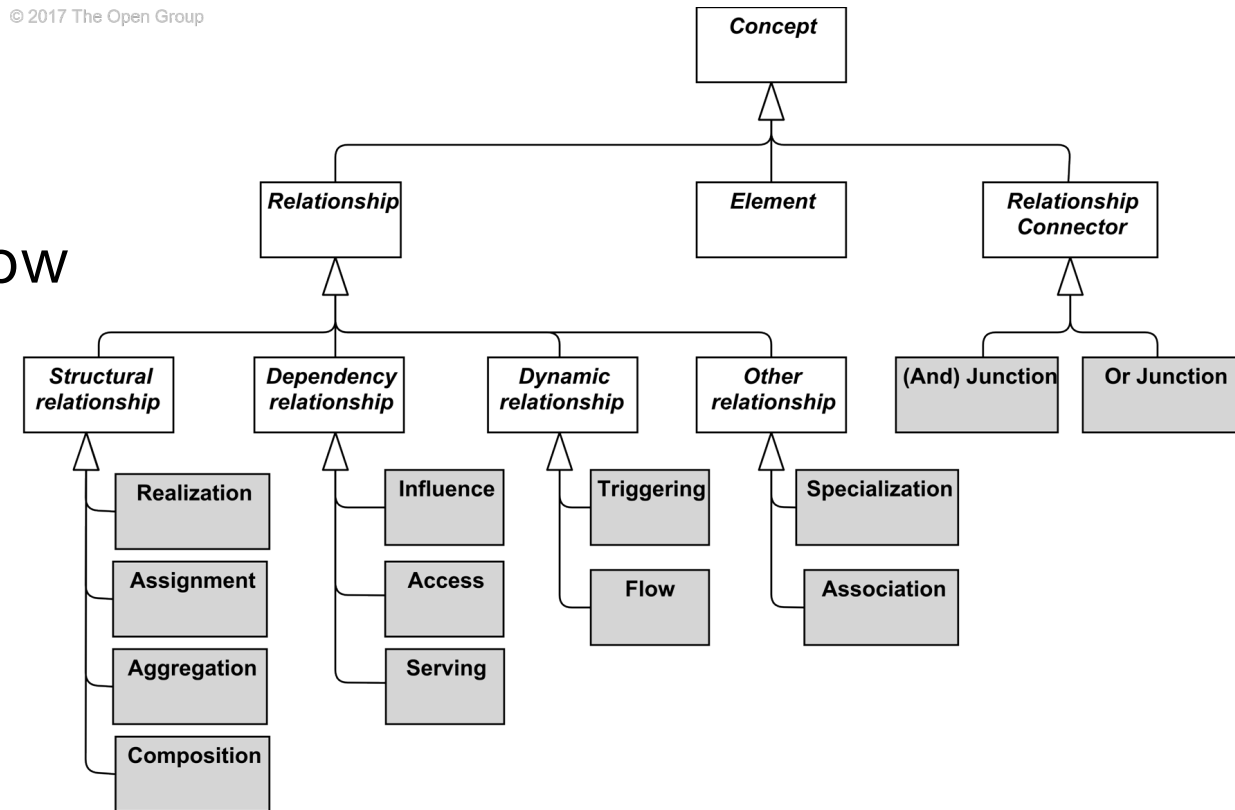
Structural Relationships

- The **composition** relationship indicates that an element consists of one or more other concepts.

- The **aggregation** relationship indicates that an element groups a number of other concepts.

- The **assignment** relationship expresses the allocation of responsibility, performance of behavior, or execution.

- The **realization** relationship indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.
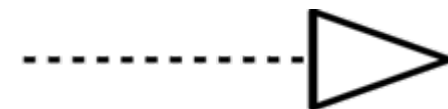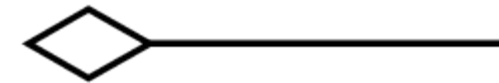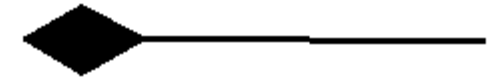
Composition relationship
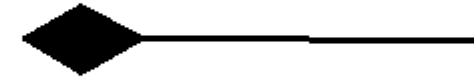
- The **composition** relationship indicates that an element **consists of** one or more other concepts.

- A composition relationship is always allowed between two instances of the same element type

© 2017 The Open Group
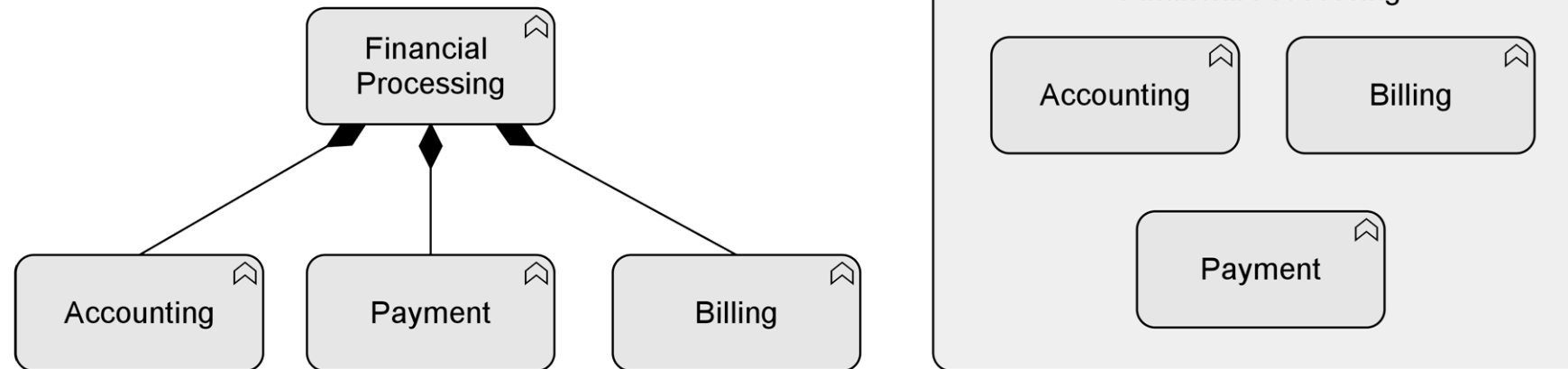
Example shows the two ways to express that the Financial Processing function is composed of three sub-functions.

Aggregation relationship

- The aggregation relationship indicates that an element **groups a number of other concepts (and they are different!!!)**
- An aggregation relationship is always allowed between two instances of the same element type
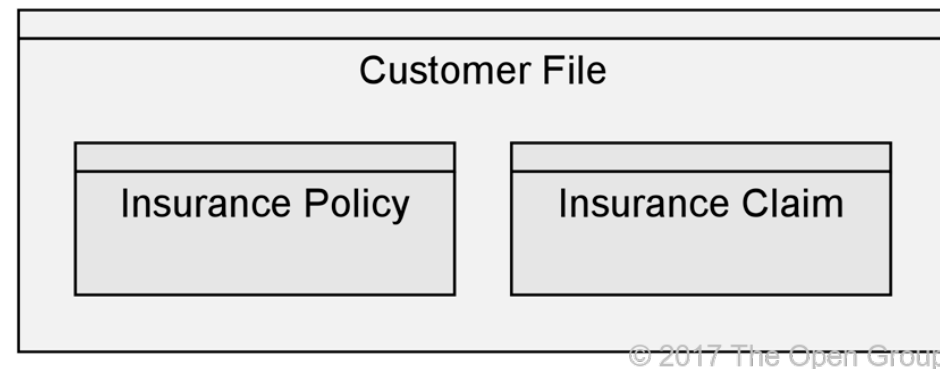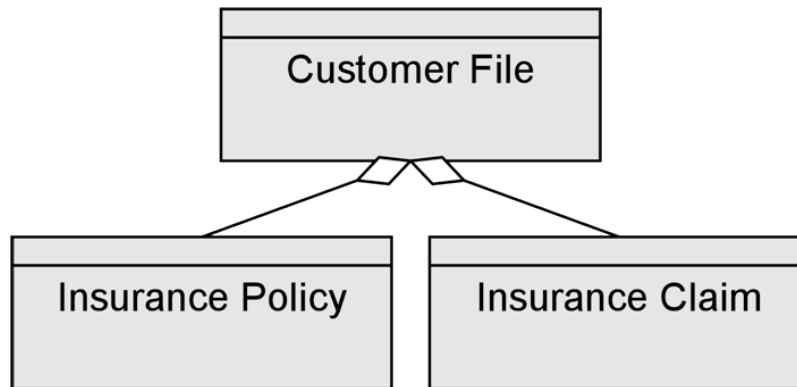
Example shows two ways to express that the Customer File aggregates an Insurance Policy and Insurance Claim.

Assignment relationship

- The assignment relationship expresses **the allocation of responsibility, performance of behavior, or execution**
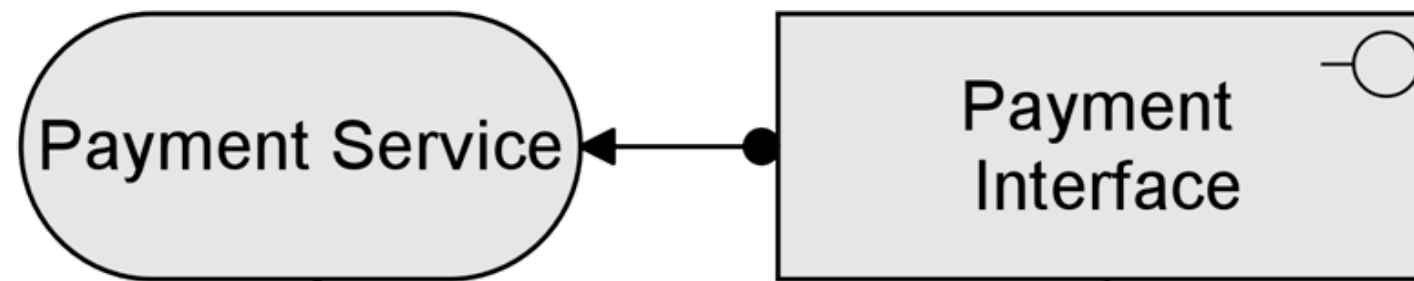
The Payment Interface is assigned to the Payment Service.

Realization relationship
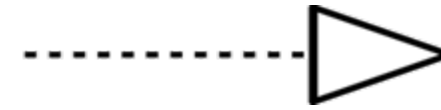
- The realization relationship indicates that an entity plays a critical role in the **creation, achievement, sustenance, or operation of a more abstract entity**

- The realization relationship indicates that more abstract entities ("what" or "logical") are realized by means of more tangible entities ("how" or "physical").

- The realization relationship is used to model run-time realization
  - E.g., that a business process realizes a business service
  - E.g., that a data object realizes a business object
  - E.g., an artifact realizes an application component
  - E.g., a core element realizes a motivation element
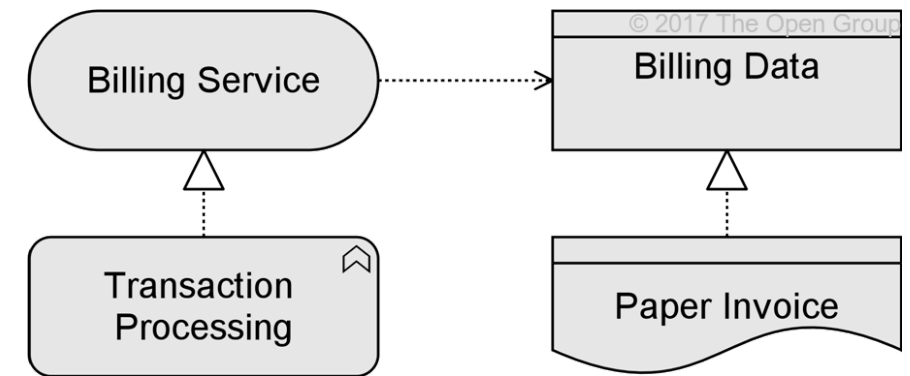
Example illustrates two ways to use the realization relationship.

A Transaction Processing function realizes a Billing Service

The Billing Data object is realized by the representation Paper Invoice.

Dependency relationships

- The *serving* relationship represents
  a *control* dependency, denoted by a solid line.

- The *access* relationship represents
  a *data* dependency, denoted by a dashed line.

- The *influence* relationship is the weakest type
  of dependency, used to model how
  motivation elements are influenced by other
  elements.
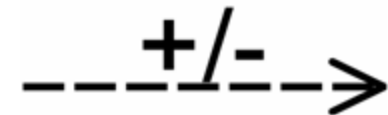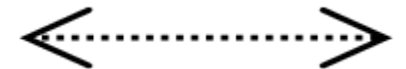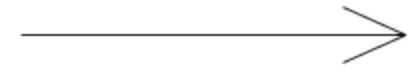
Serving

- The serving relationship models that an element provides its functionality to another element

-
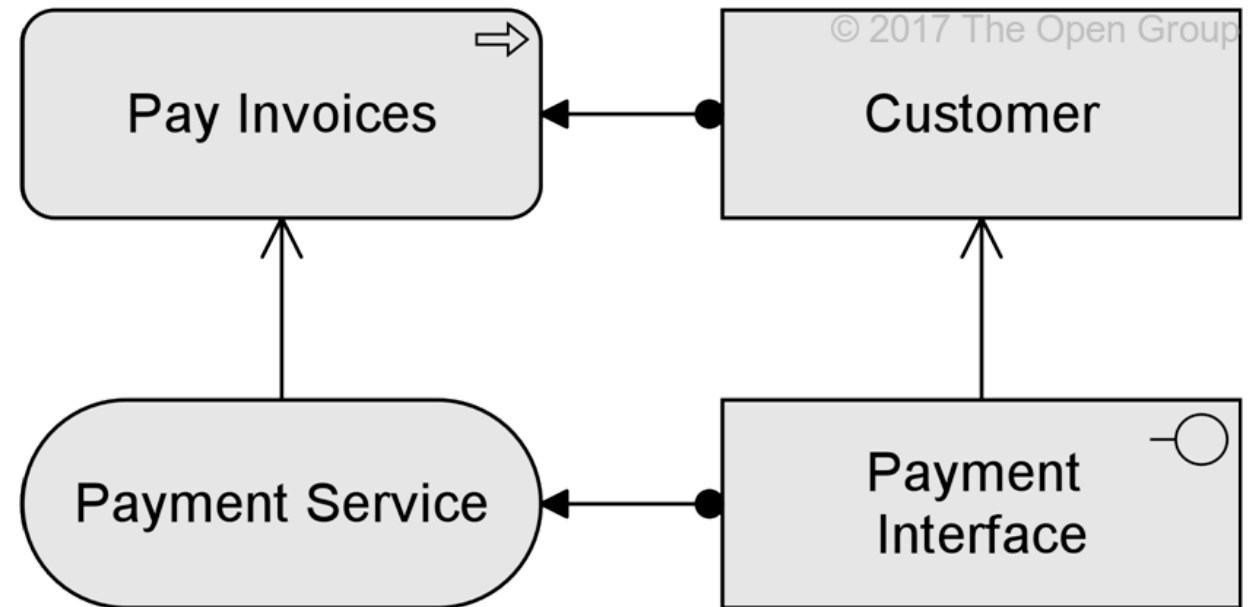
Example illustrates the serving relationship. The Payment Interface serves the Customer, while the Payment Service serves the Pay Invoices process of that customer.

Access
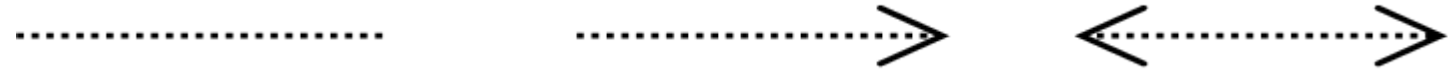
- The access relationship models the ability of behavior and active structure elements to observe or act upon passive structure elements
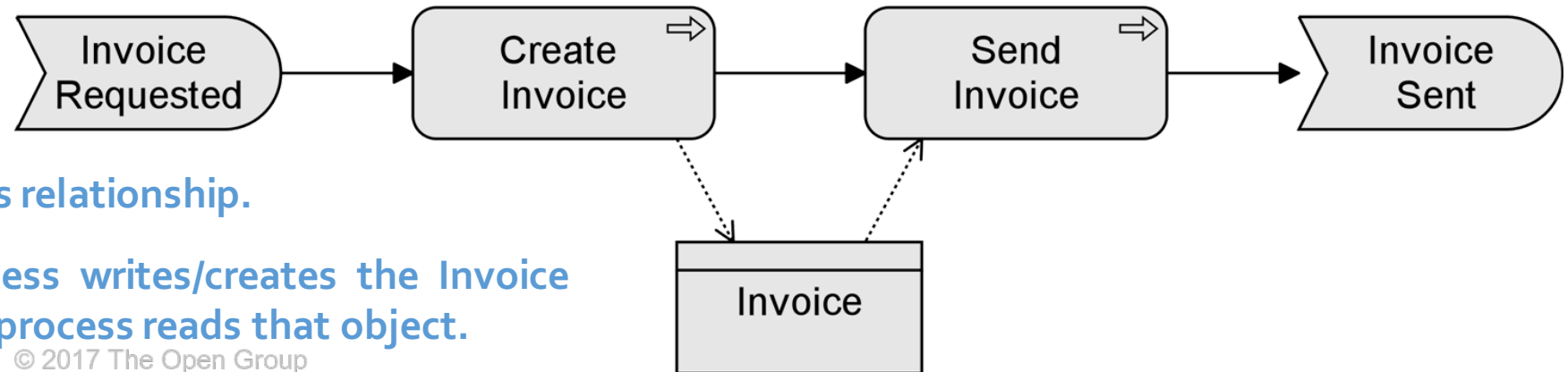
Example illustrates the access relationship.

The Create Invoice sub-process writes/creates the Invoice object; the Send Invoice sub-process reads that object.

© 2017 The Open Group

Influence
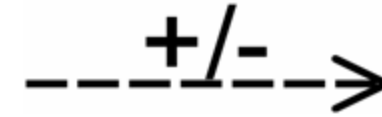
- The influence relationship models that an element affects the implementation or achievement of some motivation element
- The influence relationship is used to describe that some architectural element influences achievement or implementation of a motivation element, such as a goal or a principle

Example illustrates the use of the influence relationship to model the different effects of the same motivation element, Assign Personal Assistant. This has a strongly positive influence on Reduce Workload Of Employees, but a strongly negative influence on Decrease Costs.
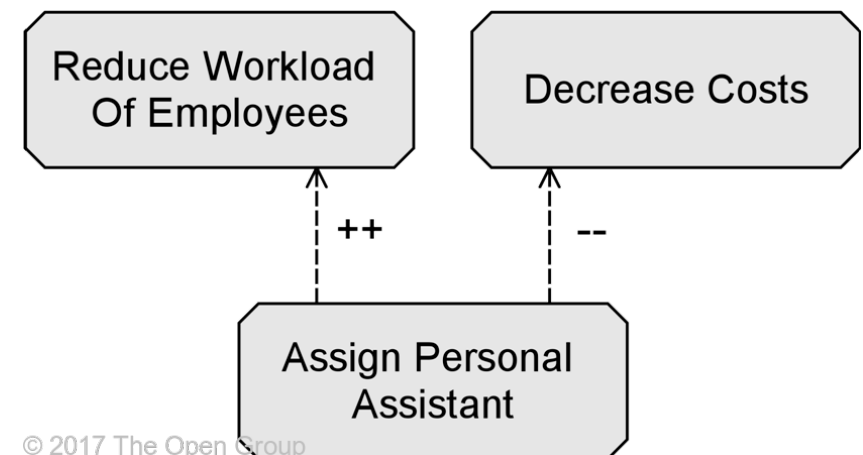
© 2017 The Open Group

Dynamic Relationships

- The **triggering relationship** describes a temporal or causal relationship between elements.

- The **flow relationship** represents transfer from one element to another.
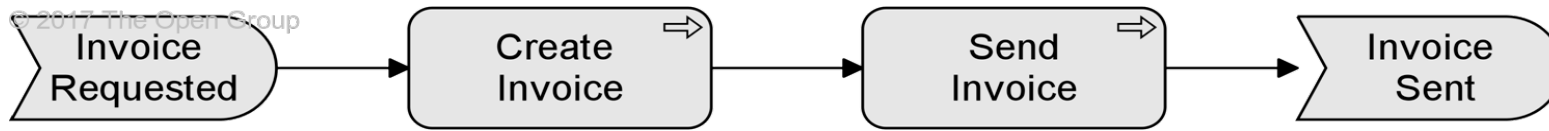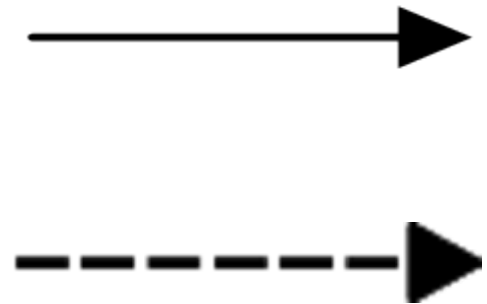
Example illustrates that triggering relationships are mostly used to model causal dependencies between (sub-)processes and/or events.
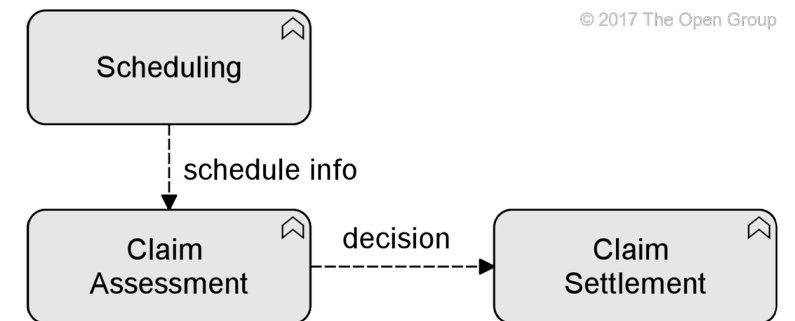
Example shows a Claim Assessment function, which forwards decisions about the claims to the Claim Settlement function. In order to determine the order in which the claims should be assessed, Claim Assessment makes use of schedule information received from the Scheduling function.

Other Relationships

- The **specialization** relationship indicates that an element is a particular kind of another element

- An **association** relationship models an unspecified relationship, or one that is not represented by another ArchiMate relationship

- A **junction** is used to connect relationships of the same type
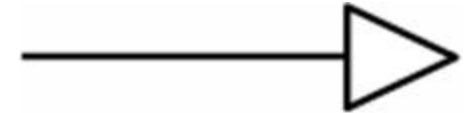
(And) Junction

Or Junction

Specialization
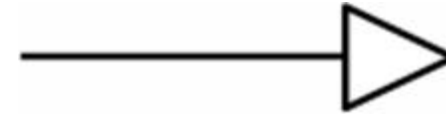
- The specialization relationship indicates that an element is a particular kind of another element.

- A specialization relationship is always allowed between two instances of the same element.

Example illustrates the use of the specialization relationship for a process. In this case the Take Out Travel Insurance and Take Out Luggage Insurance processes are a specialization of a more generic Take Out Insurance process.

© 2017 The Open Group
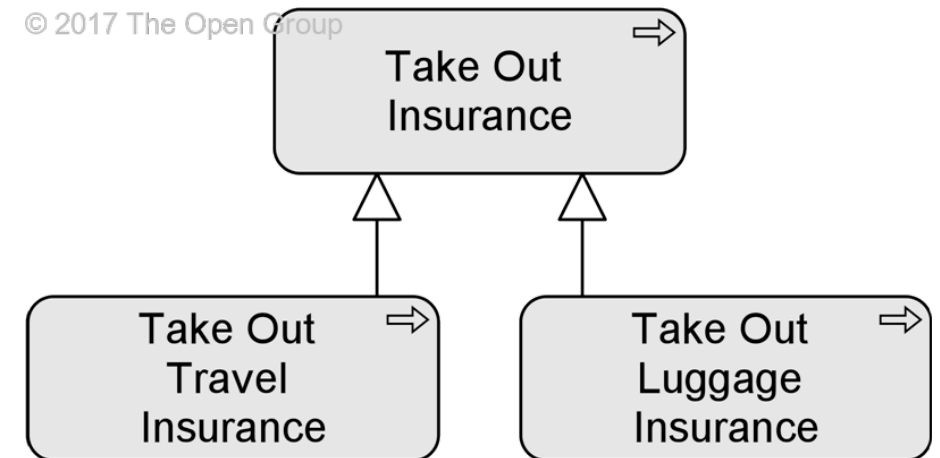
Association

- An association relationship models an unspecified relationship, or one that is not represented by another ArchiMate relationship.

- An association relationship is always allowed between two elements, or between a relationship and an element.

Example 12 illustrates a number of uses of the association relationship. It also shows an example of an association between a flow relationship and a passive structure element, to indicate the kind of information that is communicated between the two functions.

© 2017 The Open Group

Policy Creation

Policy Management

Customer File

Insurance Policy

Insured Object
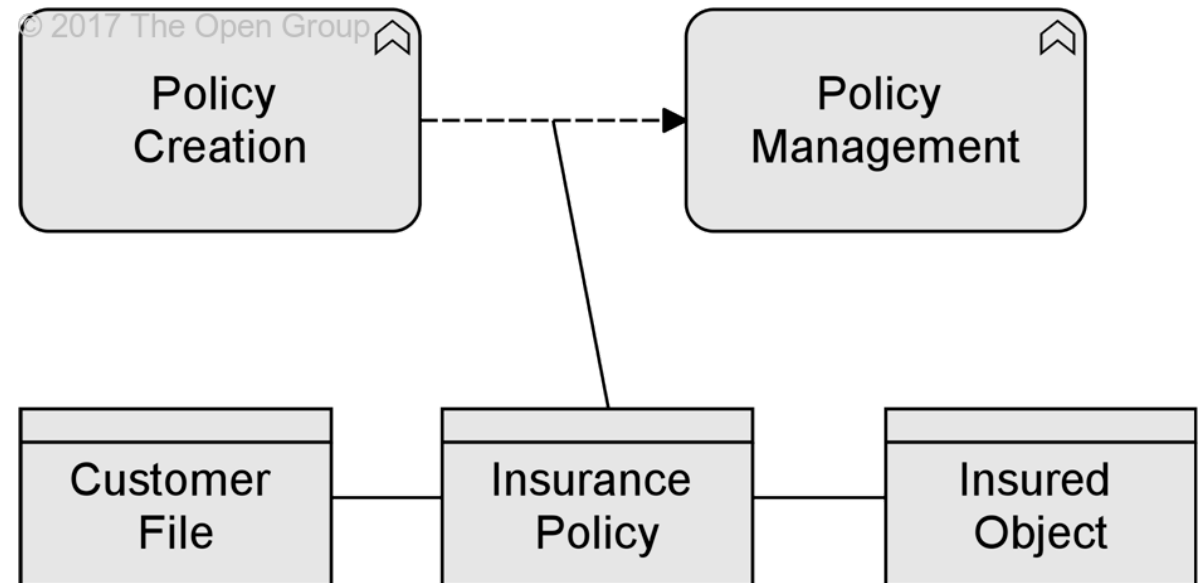
Junction

(And) Junction   **Or Junction**

- A junction is used to connect relationships of the same type.

In Example, the junction in the model is used to denote that the Sales and Finance functions together realize the Invoicing service.

In Example, the or junction is used to denote a choice: process Assess Request triggers either Accept Request or Reject Request. (The usual interpretation of two separate triggering relations, one from Assess Request to Accept Request and one from Assess Request to Reject Request, is that Assess Request triggers both of the other processes.)

Motivation Elements

Motivation Elements (part 1)

- A **stakeholder** is the role of an individual, team, or organization (or classes thereof) that represents their interests in the outcome of the architecture

- A **driver** represents an external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them

- An **assessment** represents the result of an analysis of the state of affairs of the enterprise with respect to some driver

Stakeholder
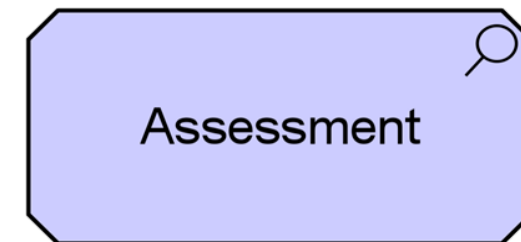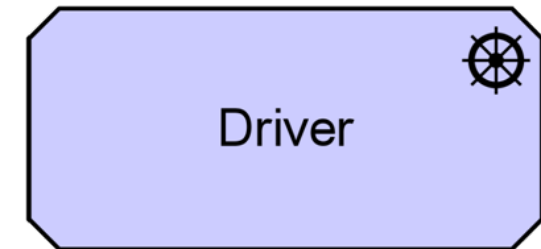
Driver

Assessment

Example - Stakeholder, Driver, and Assessment

© 2017 The Open Group
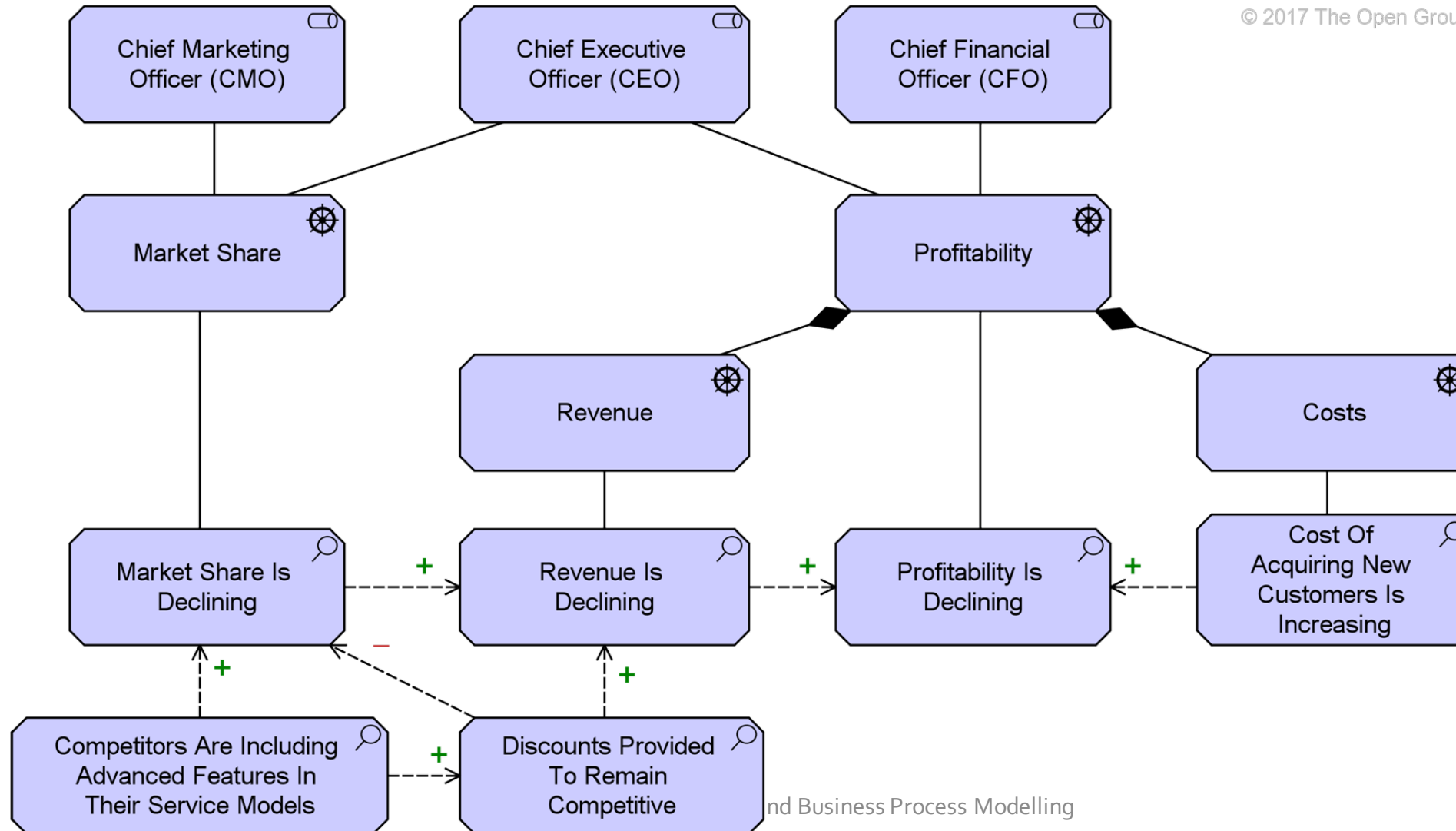
Motivation Elements (part 2)

- A **goal** represents a high-level statement of intent, direction, or desired end state for an organization and its stakeholders

- An **outcome** represents an end result that has been achieved

- A **principle** represents a qualitative statement of intent that should be met by the architecture

- A **requirement** represents a statement of need that must be met by the architecture

- A **constraint** represents a factor that prevents or obstructs the realization of goals.

Goal

Outcome

Principle

Requirement
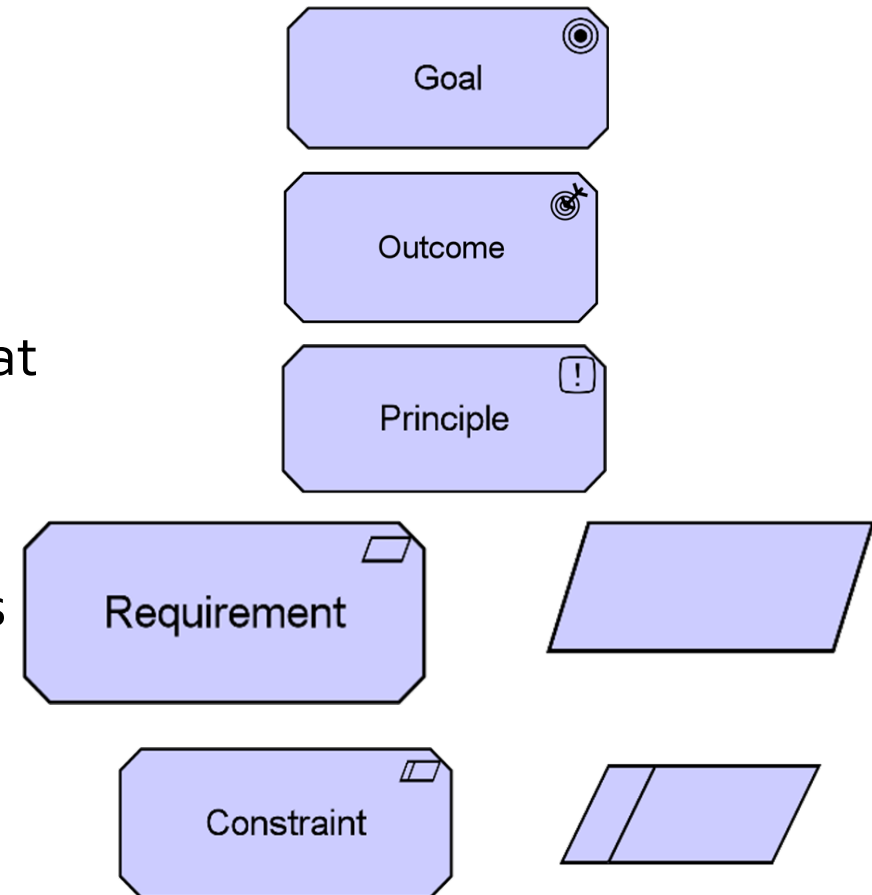
Constraint

Example: Goal, Outcome, Principle, Requirement, and Constraint
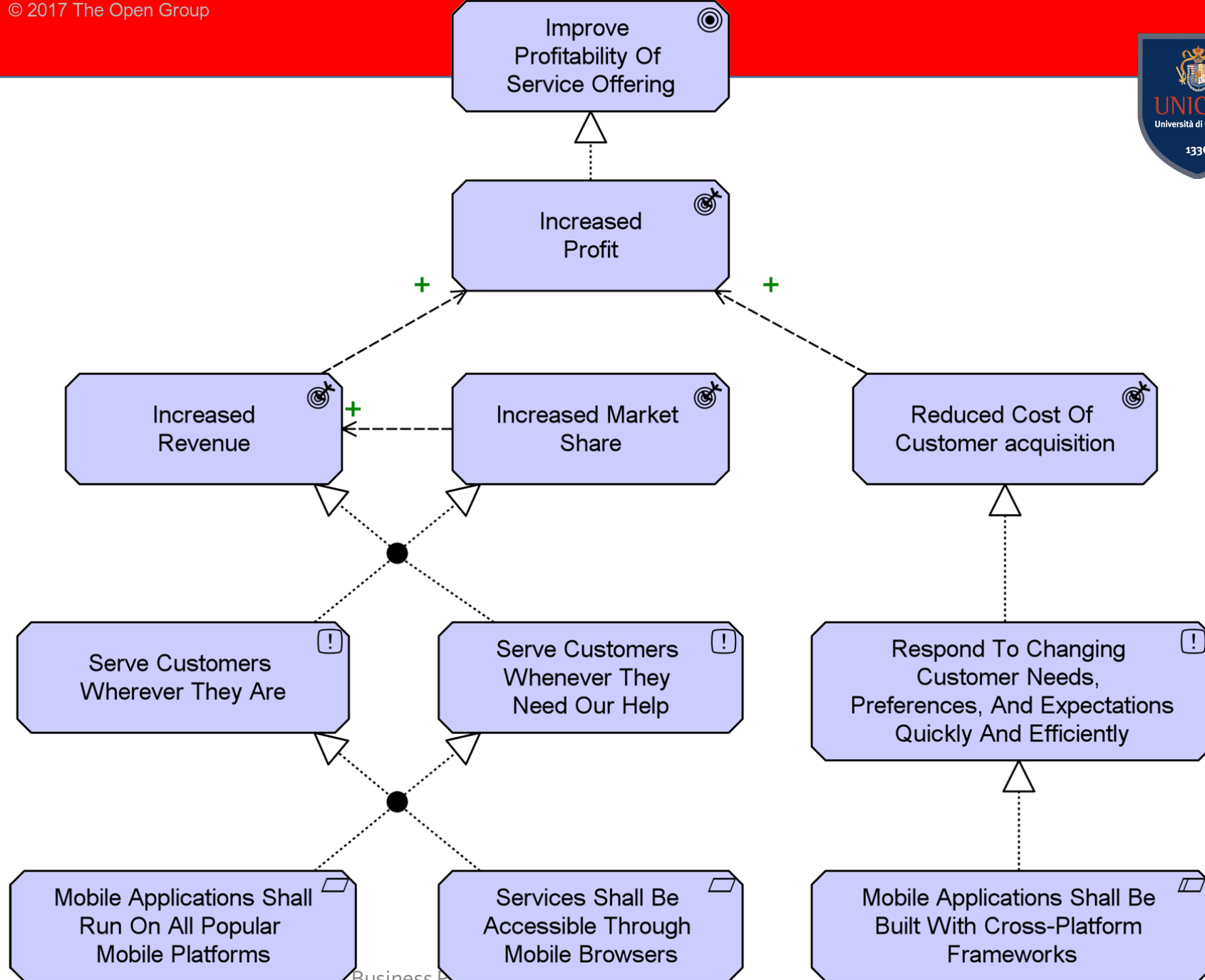
Motivation Elements (part 3)

- **Meaning** represents the knowledge or expertise present in, or the interpretation given to, a core element in a particular context

- **Value** represents the relative worth, utility, or importance of a core element or an outcome
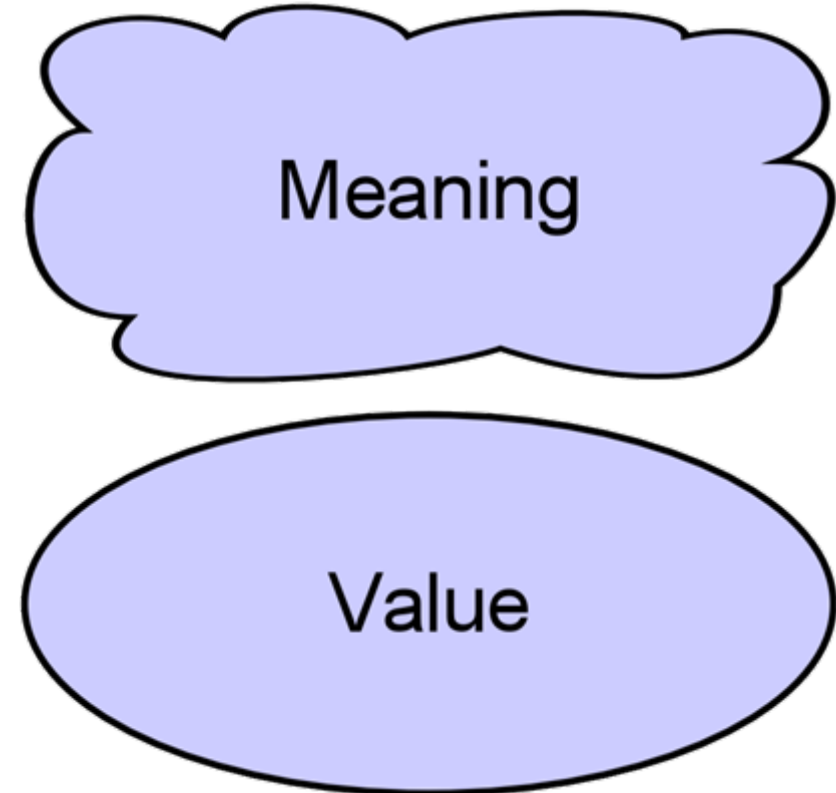
Example mining and value
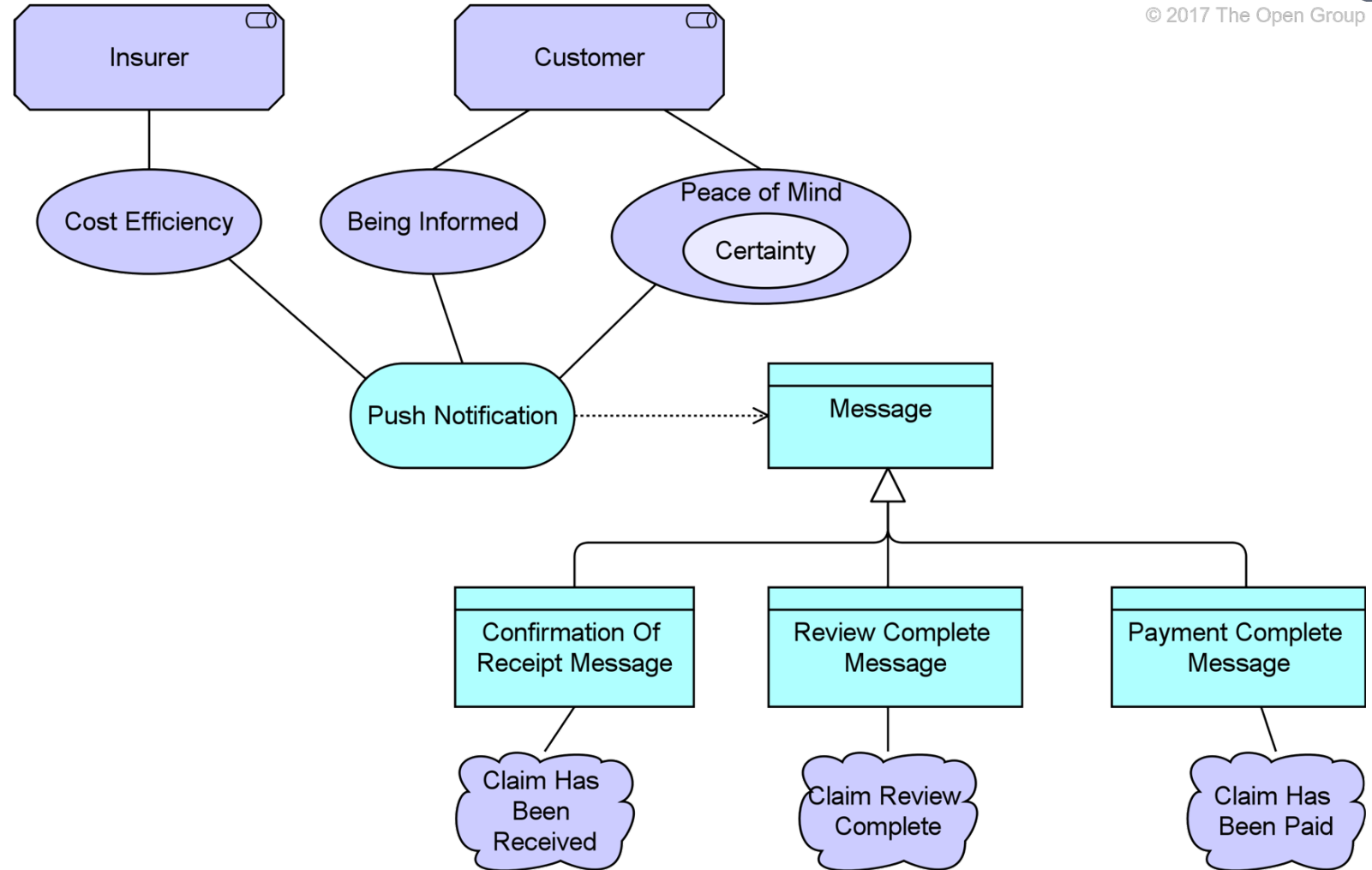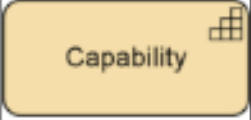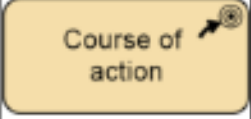
Relationships with Core Elements
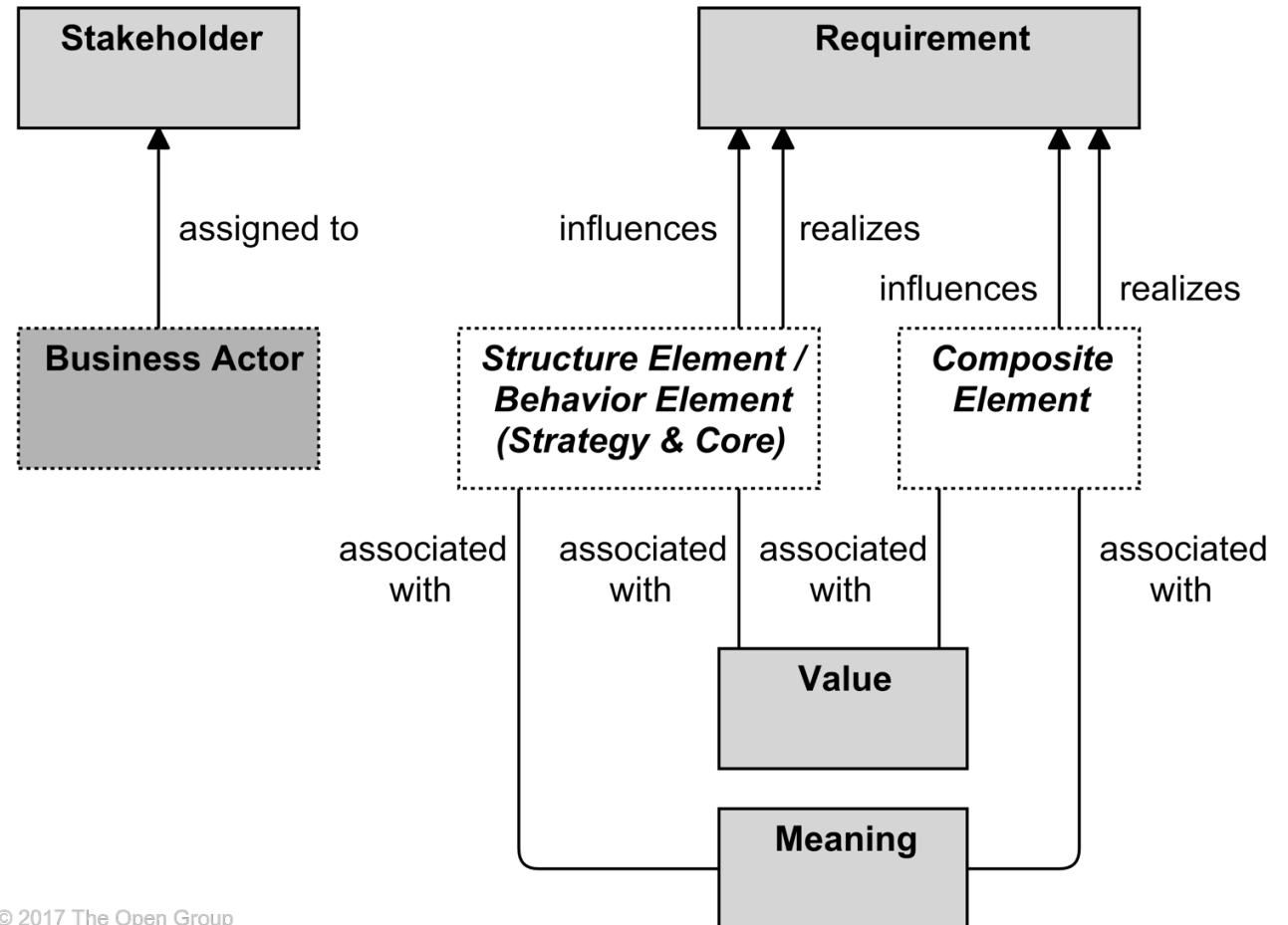
Strategy Elements

Element	Description	Notation
Resource	An asset owned or controlled by an individual or organization.	Resource
Capability	An ability that an active structure element, such as an organization, person, or system, possesses.	Capability
Course of action	An approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.	Course of action

Example Strategy Elements
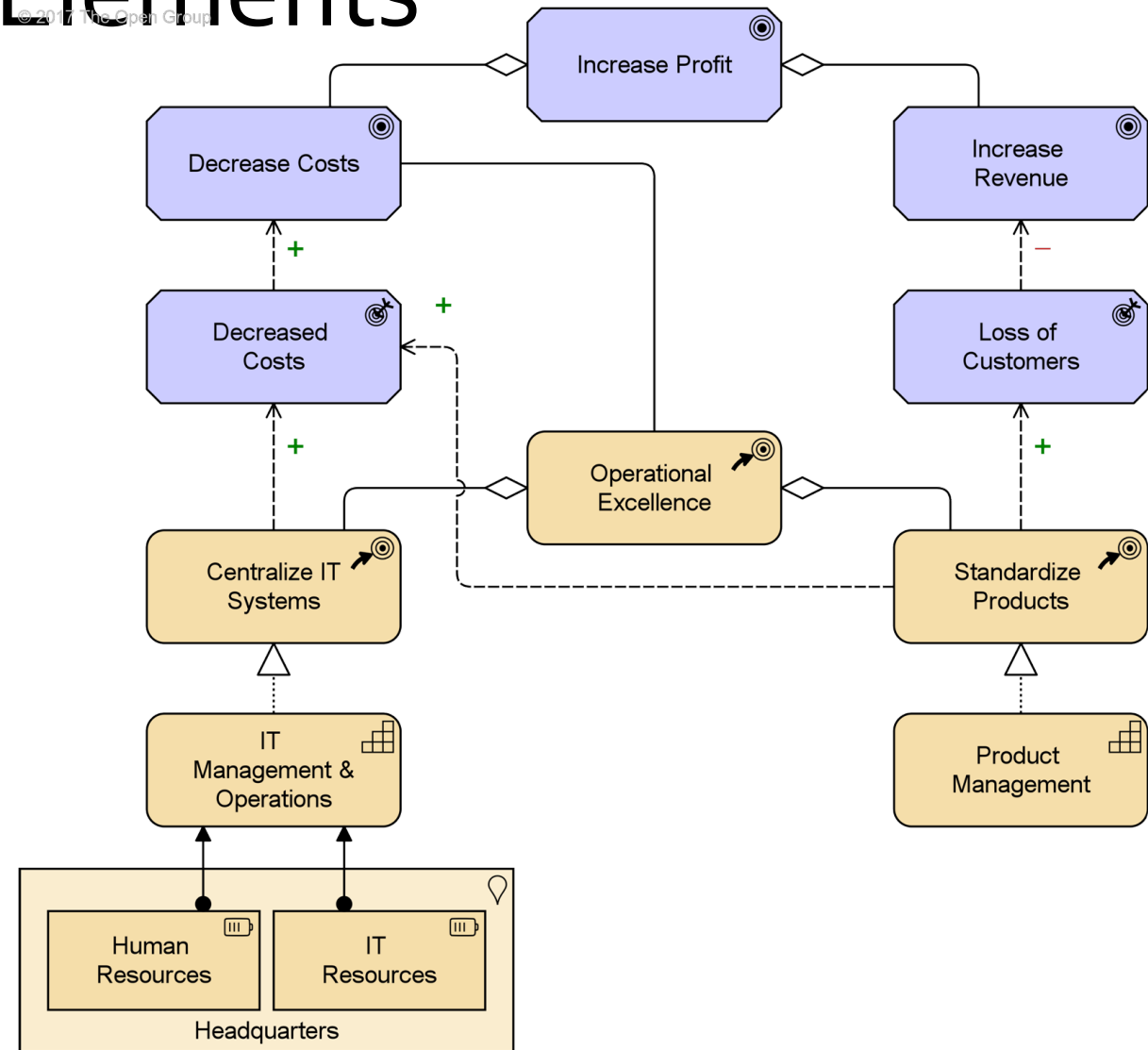
Relationships with Motivation and Core Elements

© 2017 The Open Group
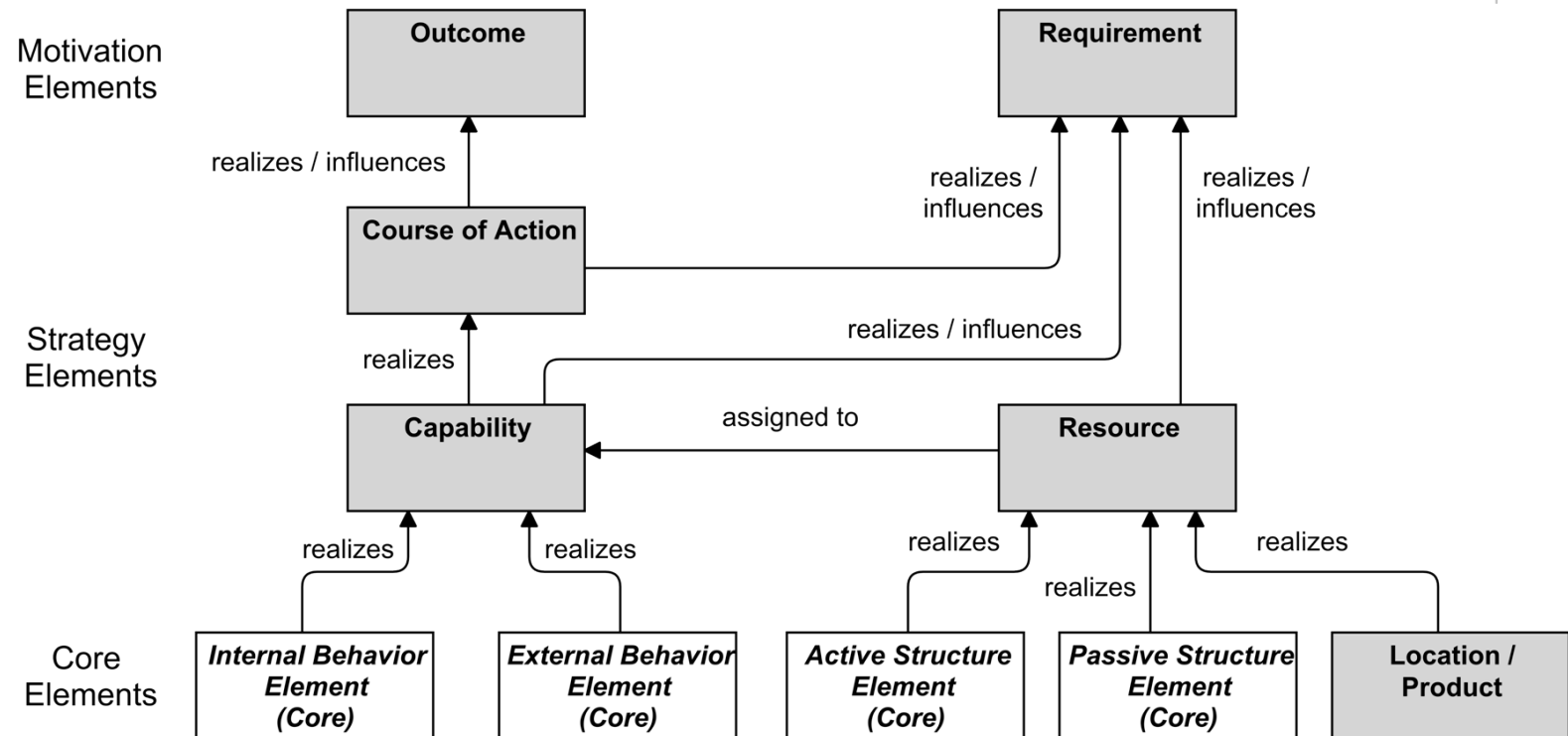
Business Layer Metamodel
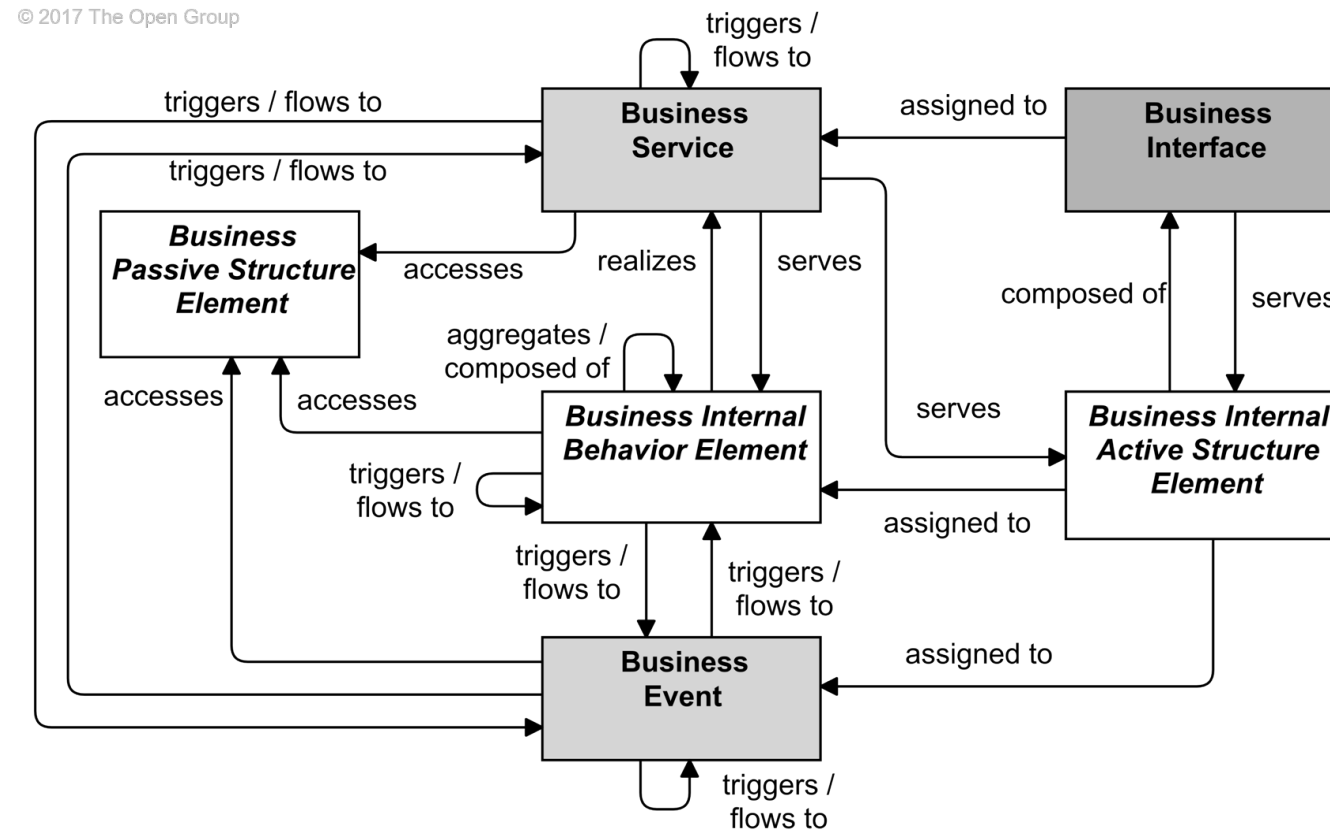
© 2017 The Open Group

Active Structure Elements
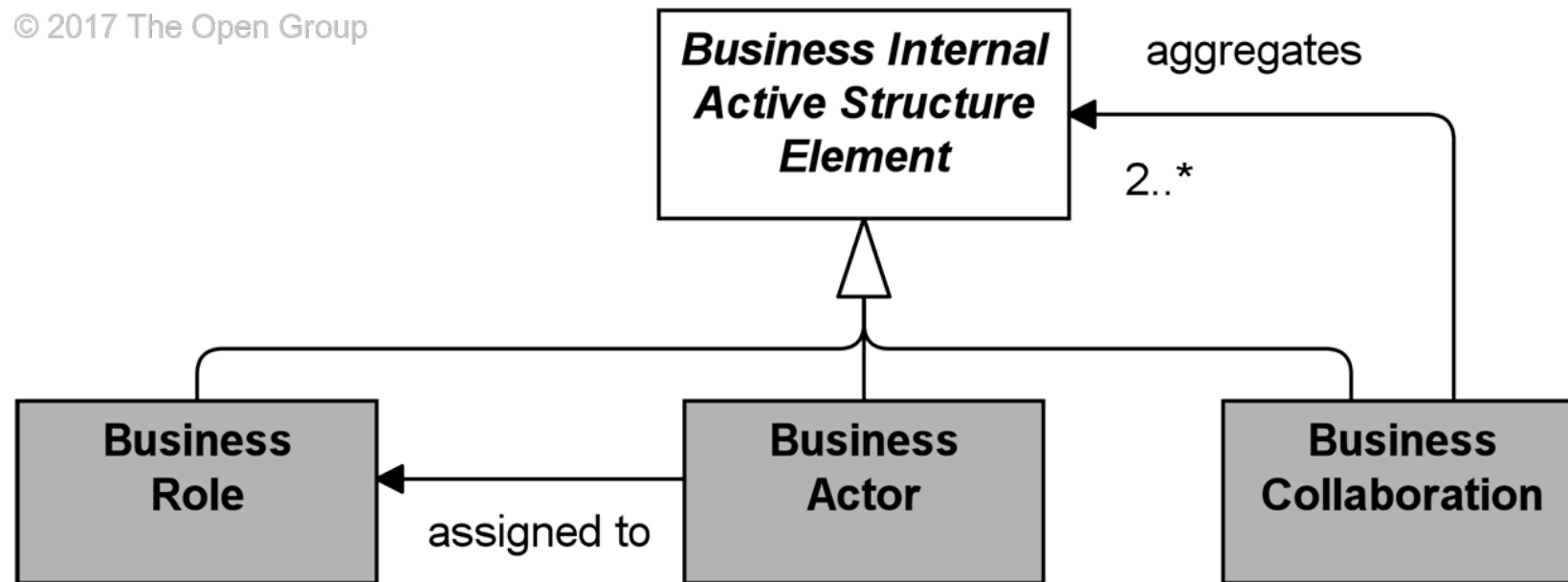
Active Structure Elements

- A **business actor** is a business entity that is capable of performing behavior

- A **business role** is the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event

- A **business collaboration** is an aggregate of two or more business internal active structure elements that work together to perform collective behavior

- A **business interface** is a point of access where a business service is made available to the environment

Example Active Structure Elements

Business Internal Behavior Elements

Business Internal Behavior Elements

- A **business process** represents a sequence of business behaviors that achieves a specific outcome such as a defined set of products or business services.

- A **business function** is a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization.

- A **business interaction** is a unit of collective business behavior performed by (a collaboration of) two or more business roles.

- A **business event** is a business behavior element that denotes an organizational state change. It may originate from and be resolved inside or outside the organization.

- A **business service** represents an explicitly defined exposed business behavior.
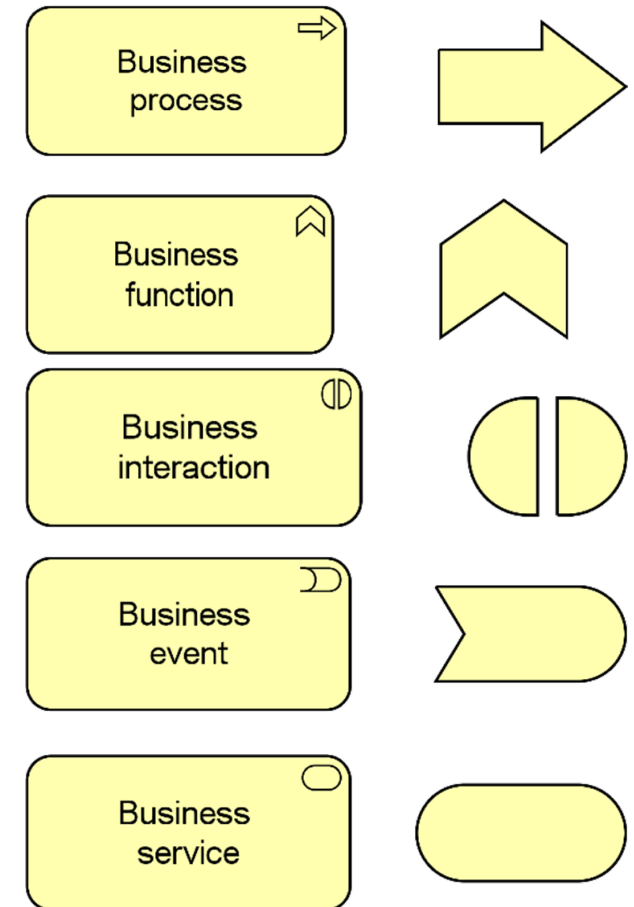
Example Business Internal Behavior Elements
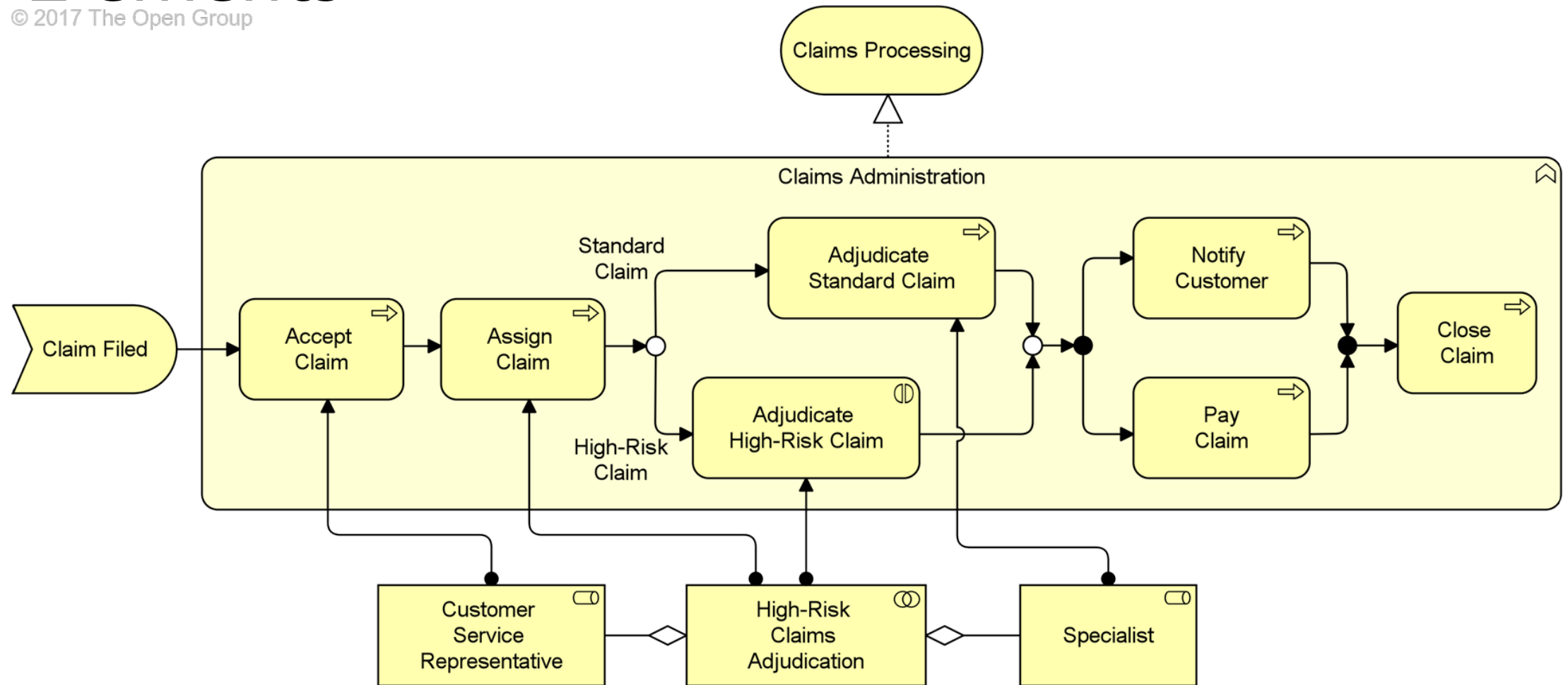
Passive structure aspect
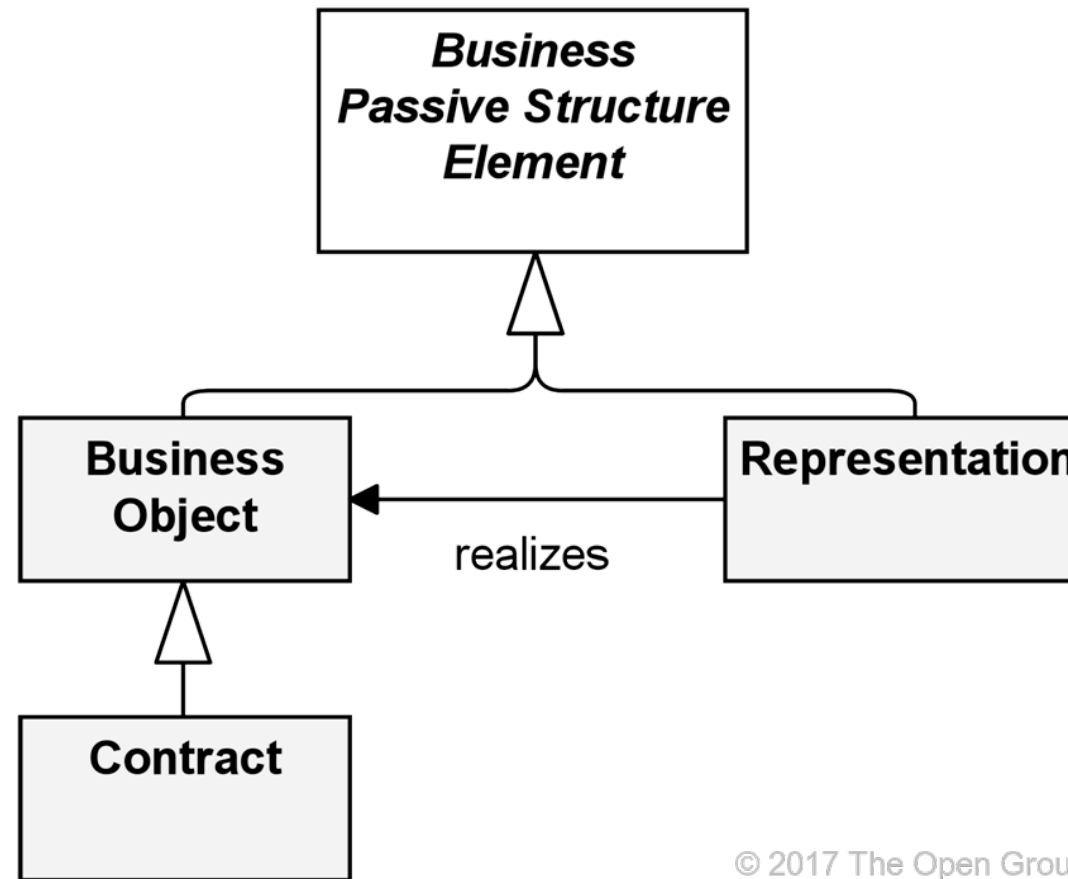
© 2017 The Open Group

# Passive structure aspect

- A **business object** represents a concept used within a particular business domain

- A **contract** represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction

- A **representation** represents a perceptible form of the information carried by a business object
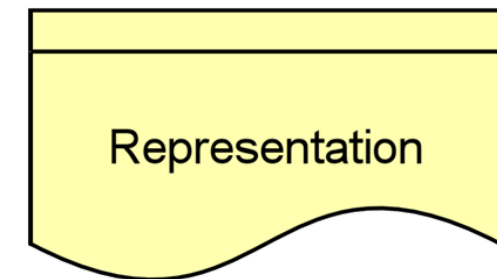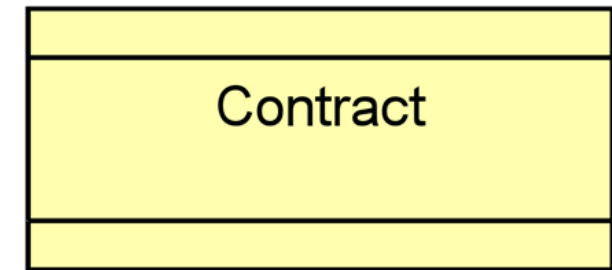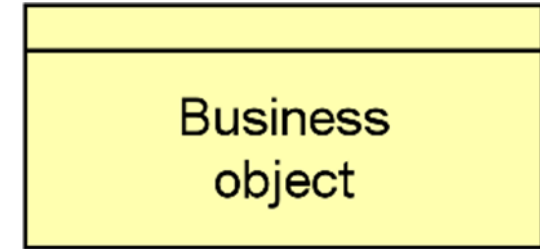
Example Passive structure aspect

Application Layer Metamodel
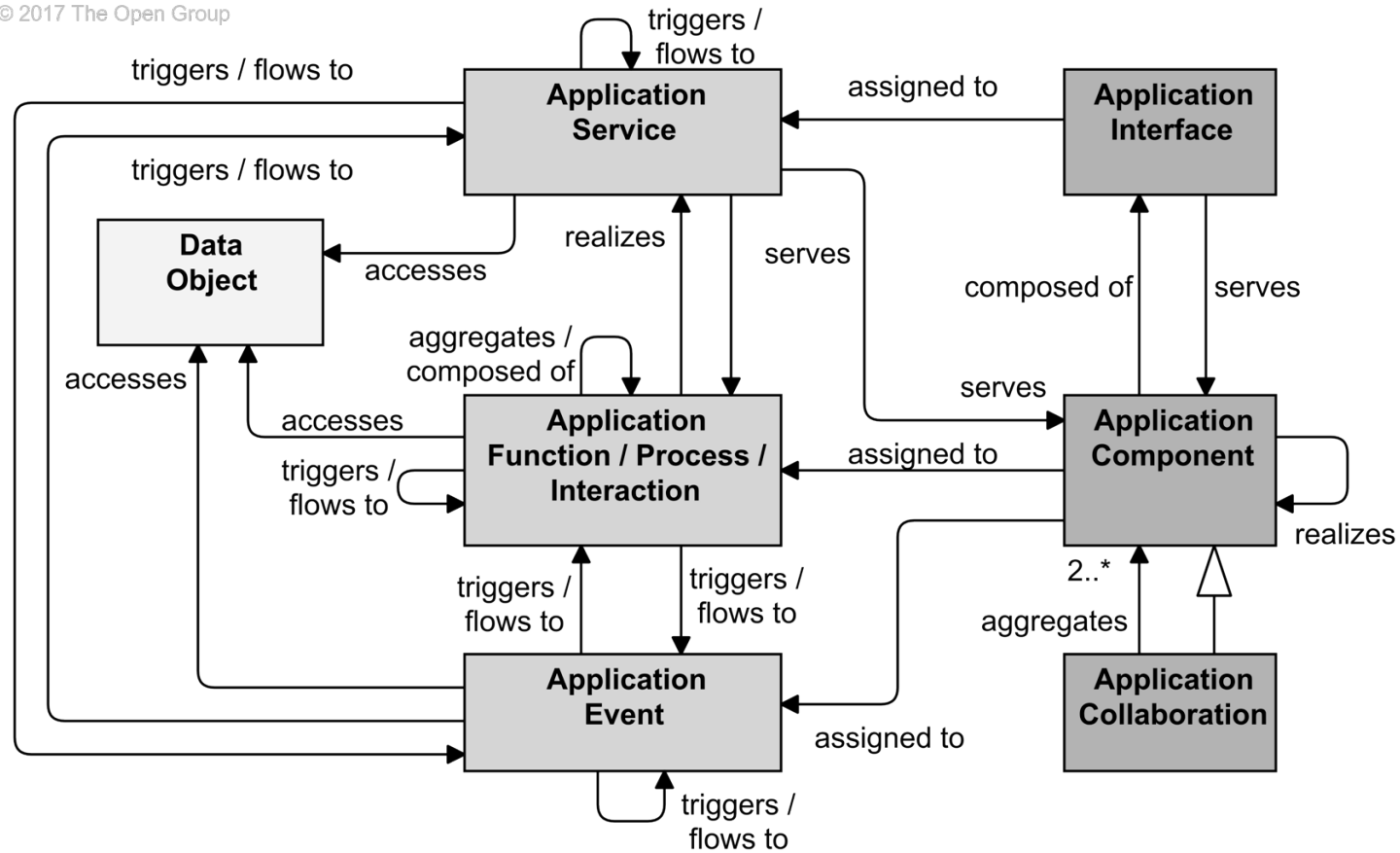
Active Structure Elements

- An **application component** represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. It encapsulates its behavior and data, exposes services, and makes them available through interfaces

- An **application collaboration** represents an aggregate of two or more application components that work together to perform collective application behavior

- An **application interface** represents a point of access where application services are made available to a user, another application component, or a node
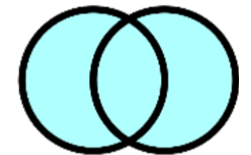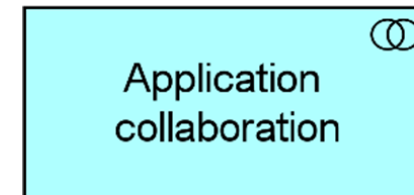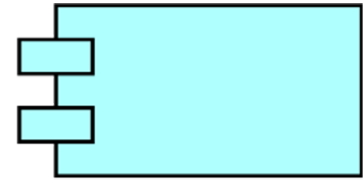
Example - Active Structure Elements
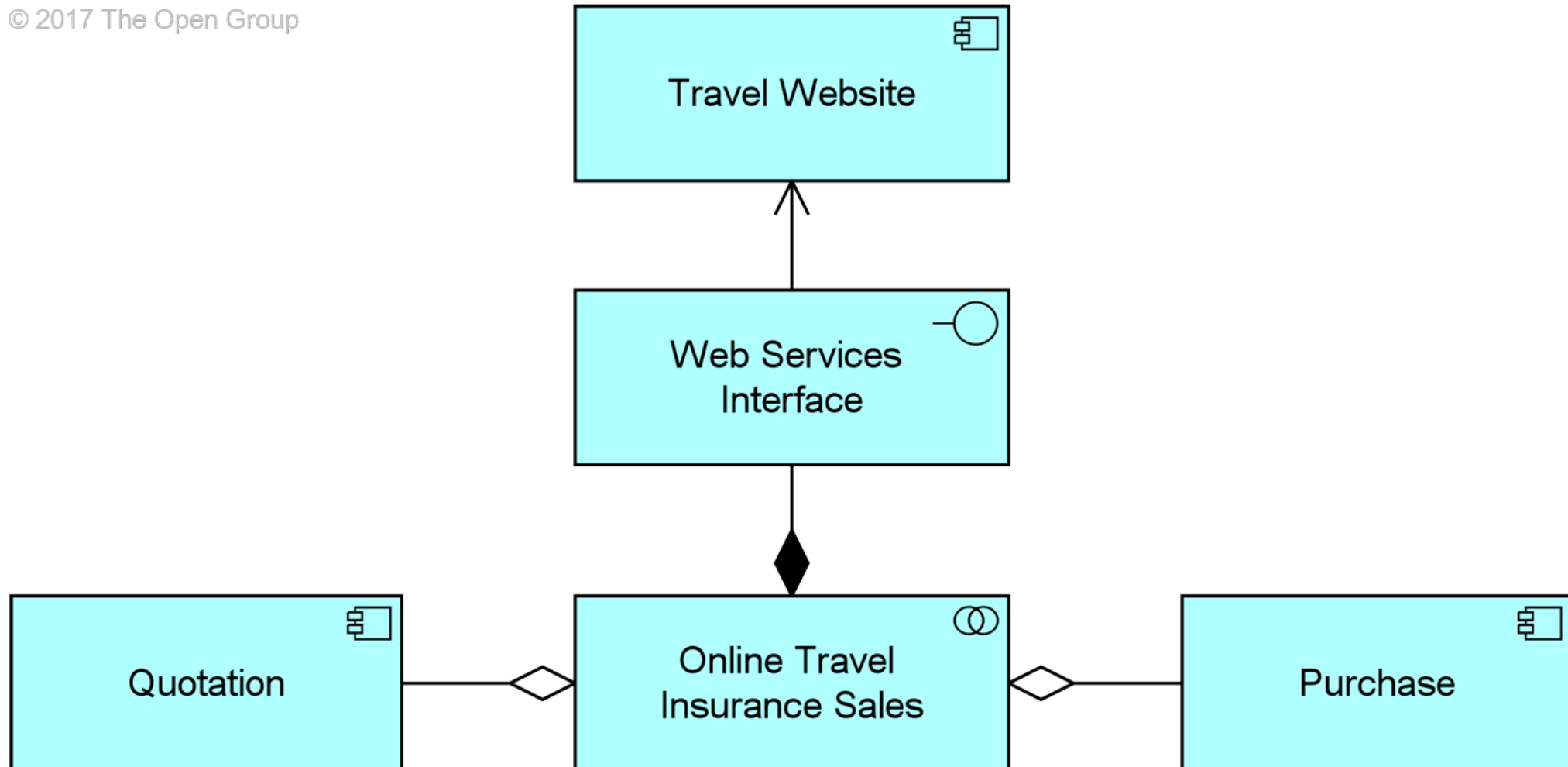
Behavior Elements

- An **application function** represents automated behavior that can be performed by an application component.

- An **application interaction** represents a unit of collective application behavior performed by (a collaboration of) two or more application components.

- An **application process** represents a sequence of application behaviors that achieves a specific outcome.

- An **application event** is an application behavior element that denotes a state change.

- An **application service** represents an explicitly defined exposed application behavior.

Application function

Application interaction

Application process

Application event

Application service
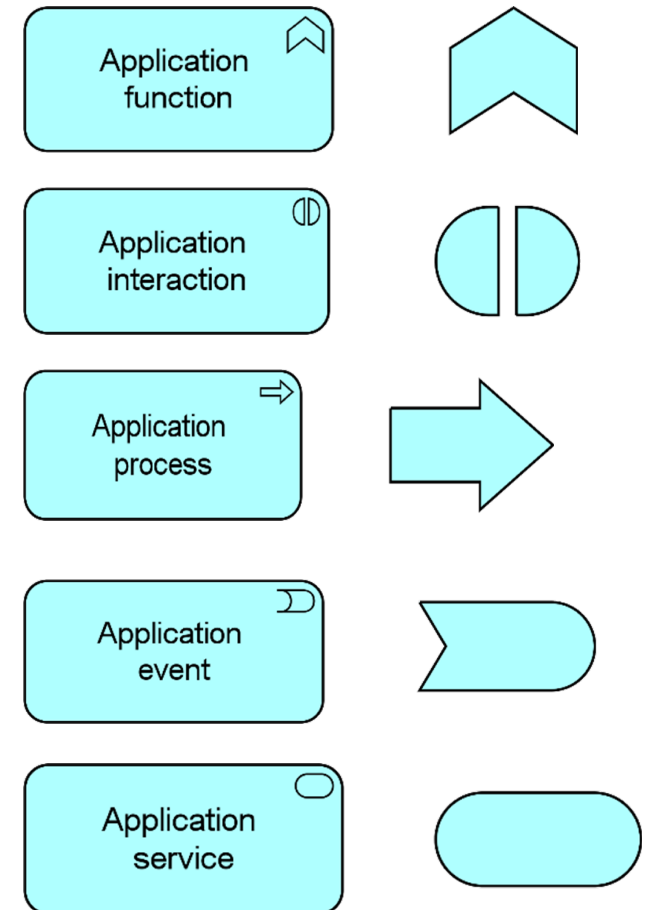
Example Behavior Elements
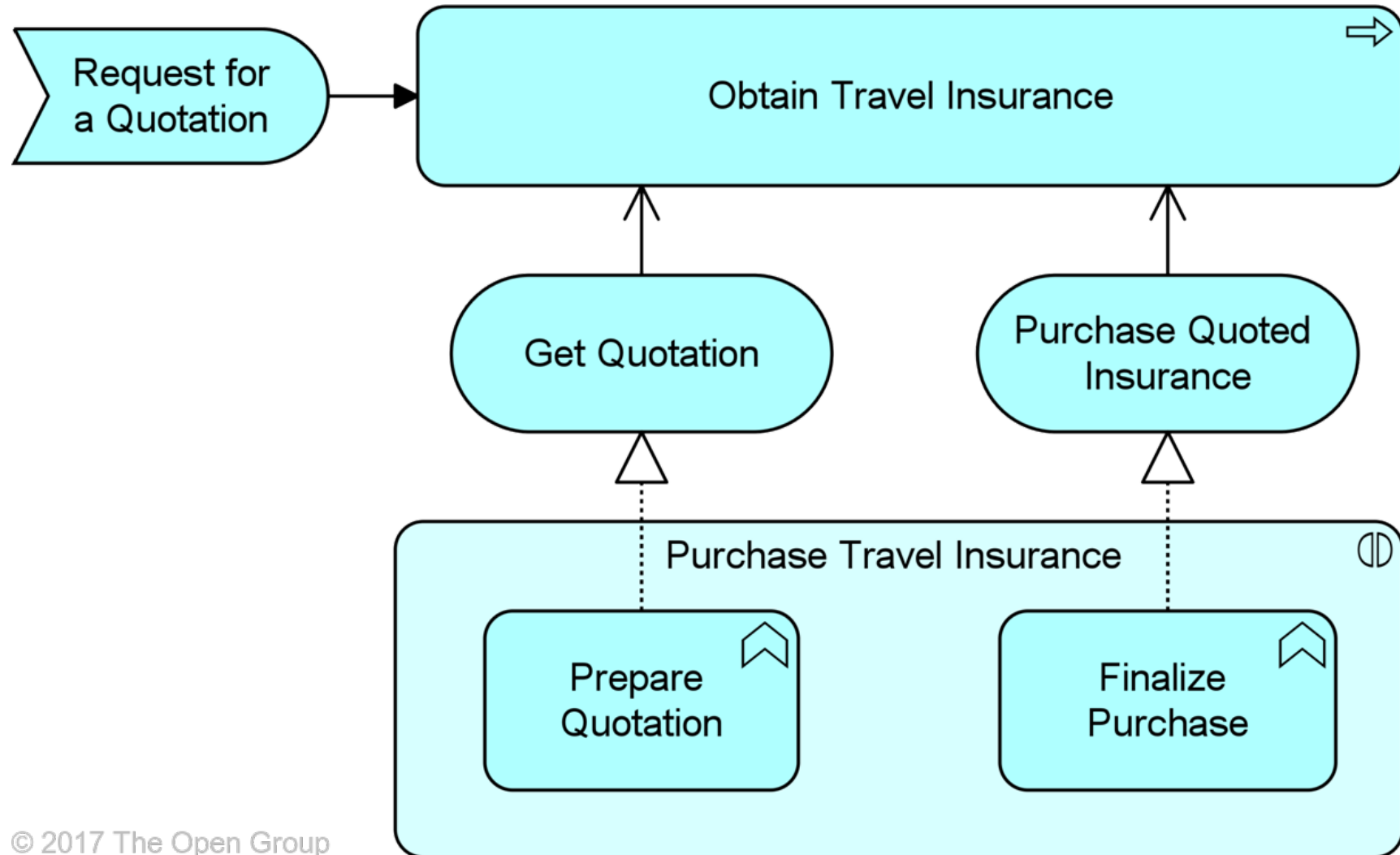
Passive Structure Elements

- A data object represents data structured for automated processing.
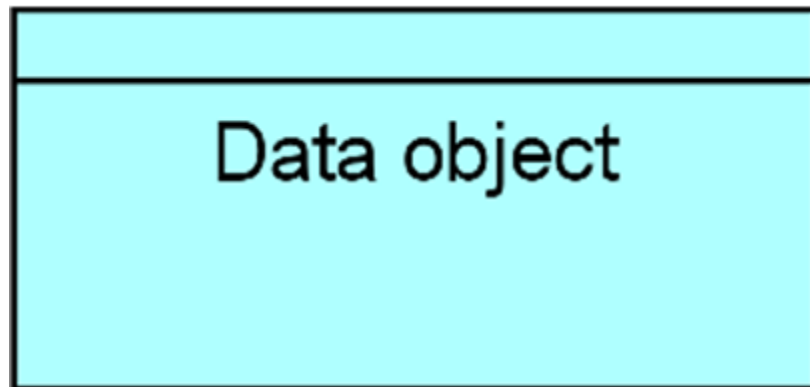
© 2017 The Open Group
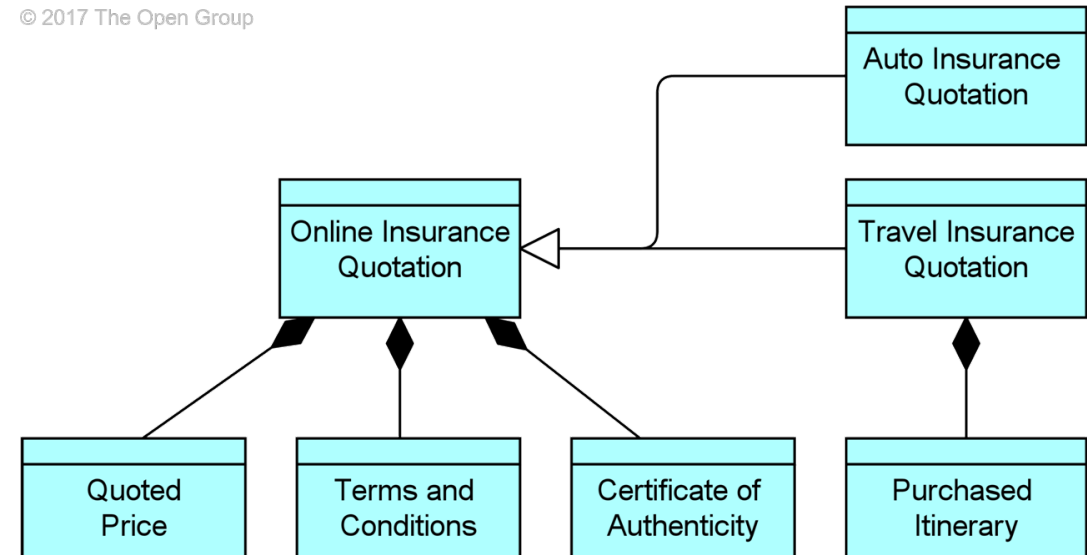
Technology layer
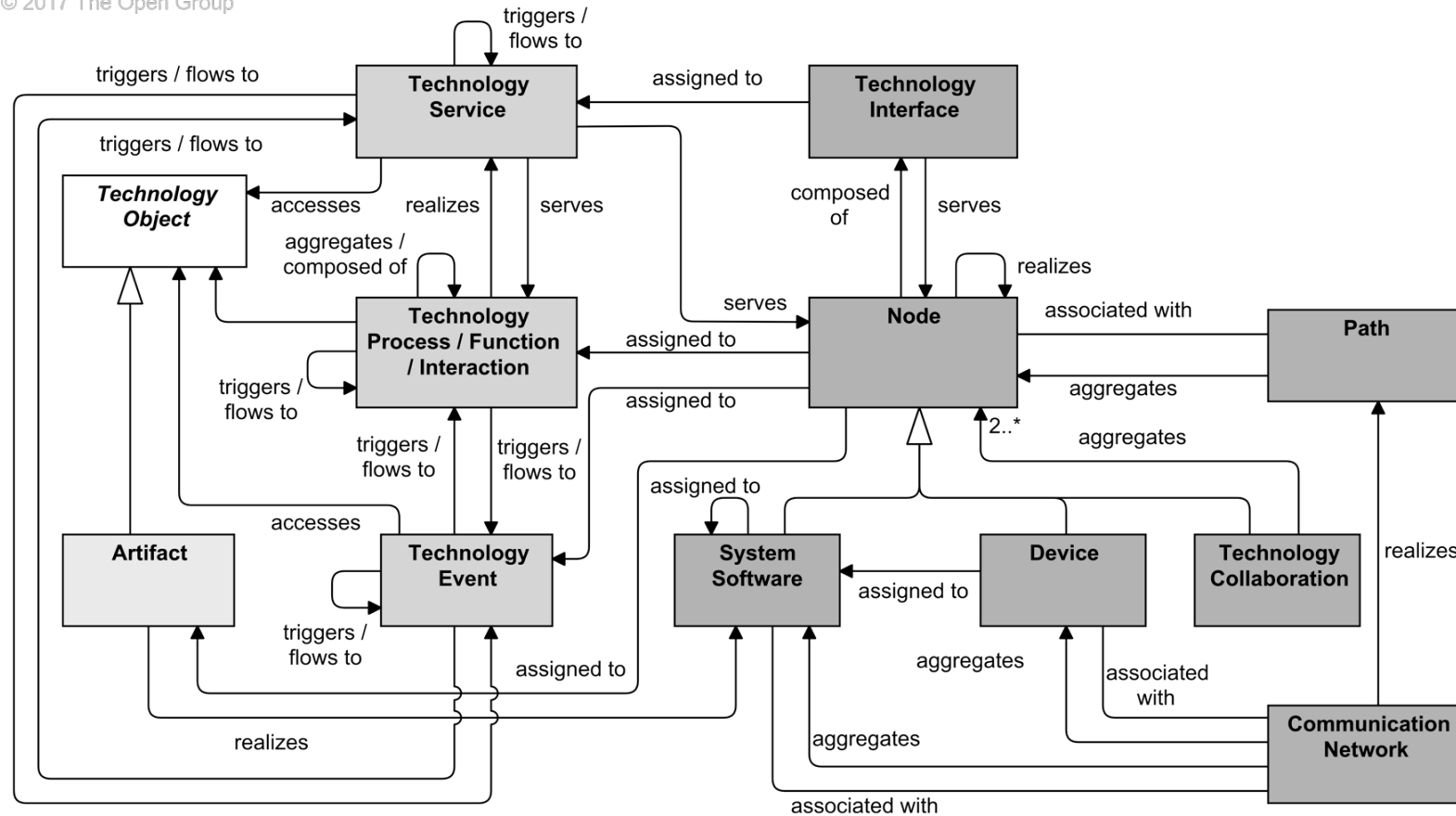
Active Structure Elements

- A node represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.

- A device is a physical IT resource upon which system software and artifacts may be stored or deployed for execution.

- System software represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.

- A technology collaboration represents an aggregate of two or more nodes that work together to perform collective technology behavior.

- A technology interface represents a point of access where technology services offered by a node can be accessed.

- A path represents a link between two or more nodes, through which these nodes can exchange data or material

- A communication network represents a set of structures that connects computer systems or other electronic devices for transmission, routing, and reception of data or data-based communications such as voice and video

Node

Device

System software

Technology collaboration

Technology interface

Path

Communication Network
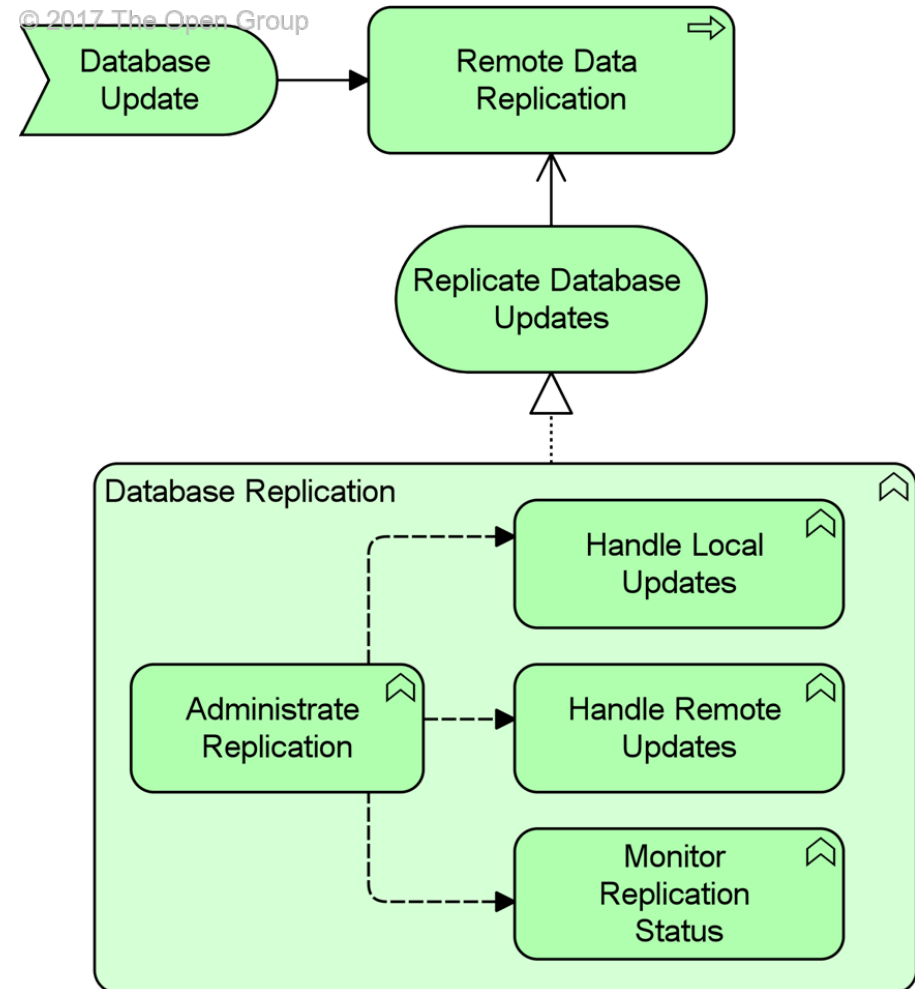
Example Active Structure Elements

65
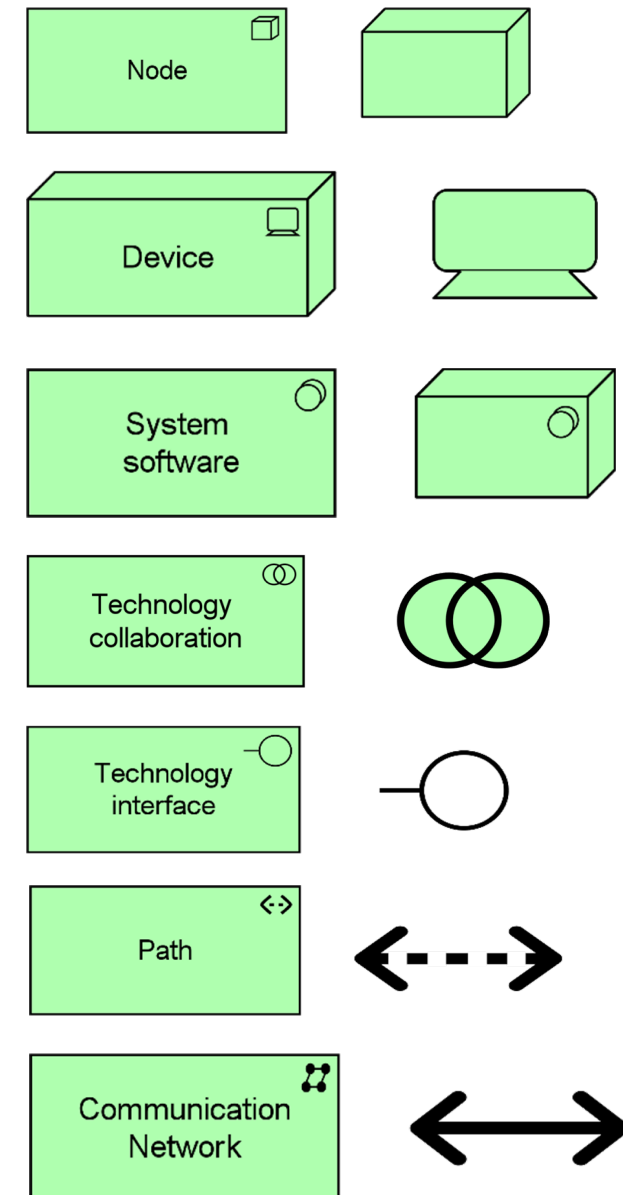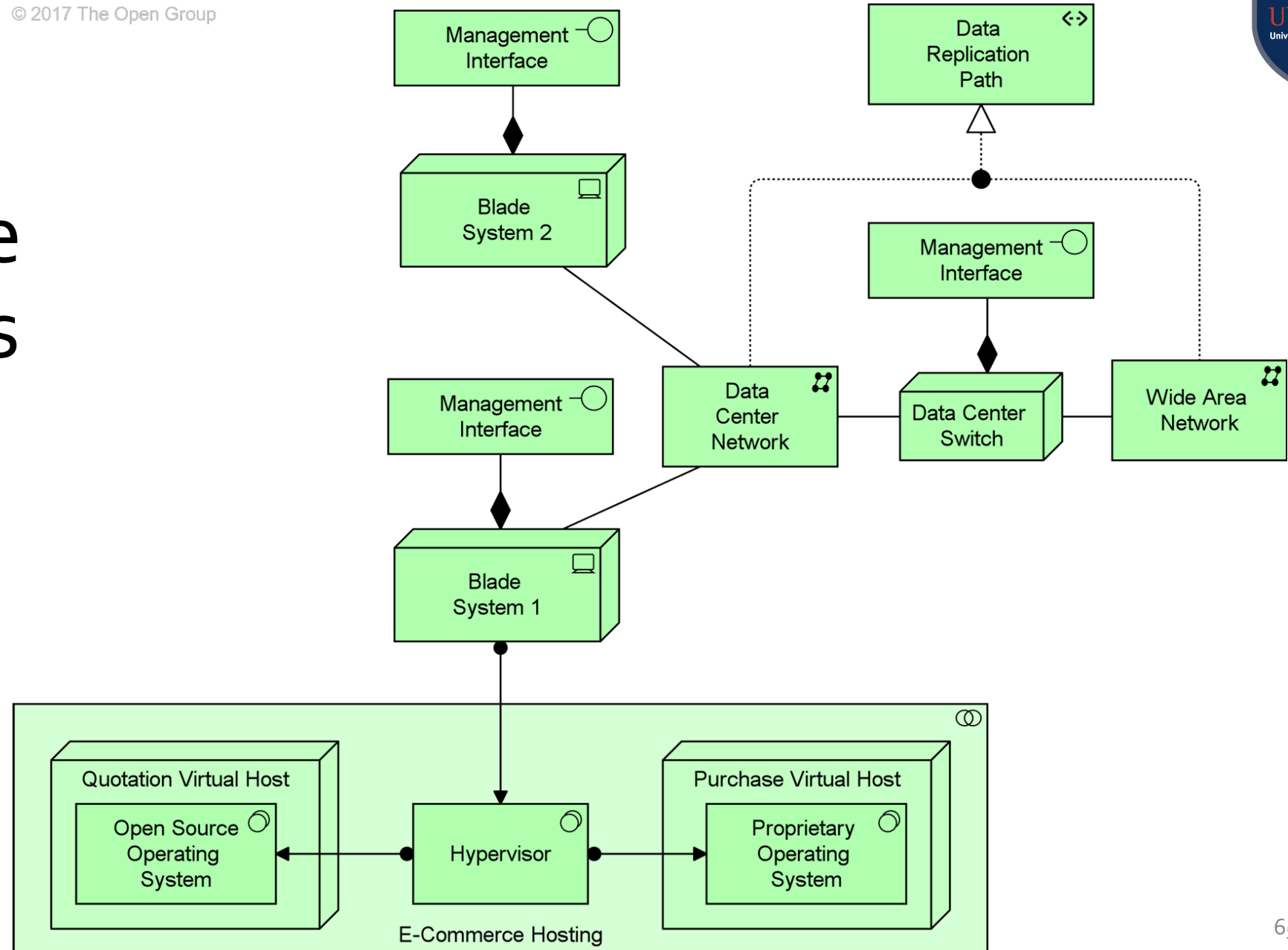
Behavioral Elements

- A technology function represents a collection of technology behavior that can be performed by a node.

- A technology process represents a sequence of technology behaviors that achieves a specific outcome.

- A technology interaction represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.

- A technology event is a technology behavior element that denotes a state change.

- A technology service represents an explicitly defined exposed technology behavior.

Technology function

Technology process

Technology interaction

Technology event

Technology service
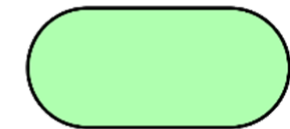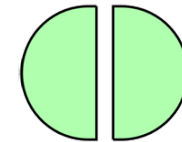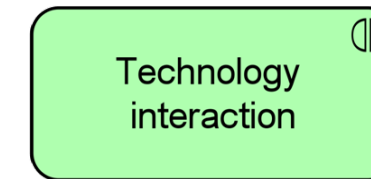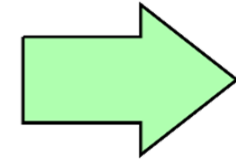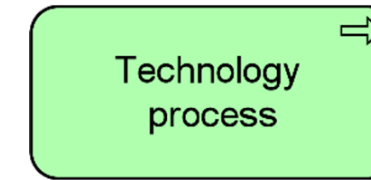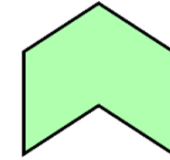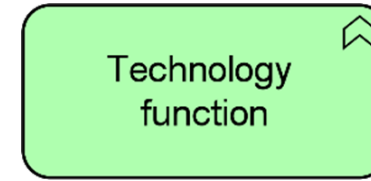
Example Behavioral Elements

Passive Structure Elements

- An artifact represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system
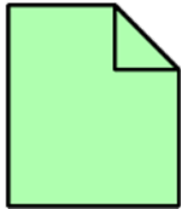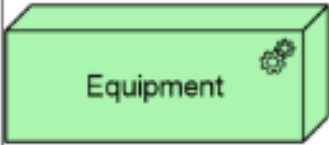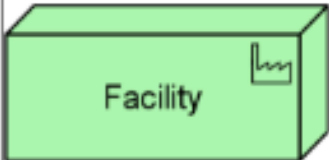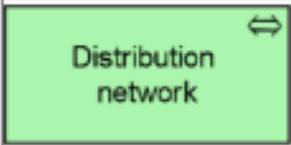

© 2017 The Open Group

Physical elements

© 2017 The Open Group
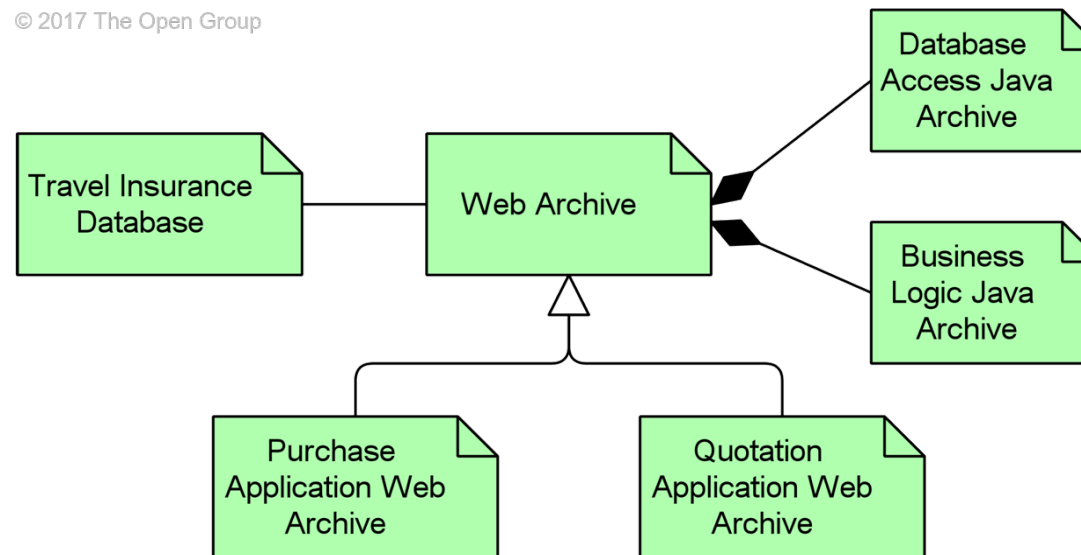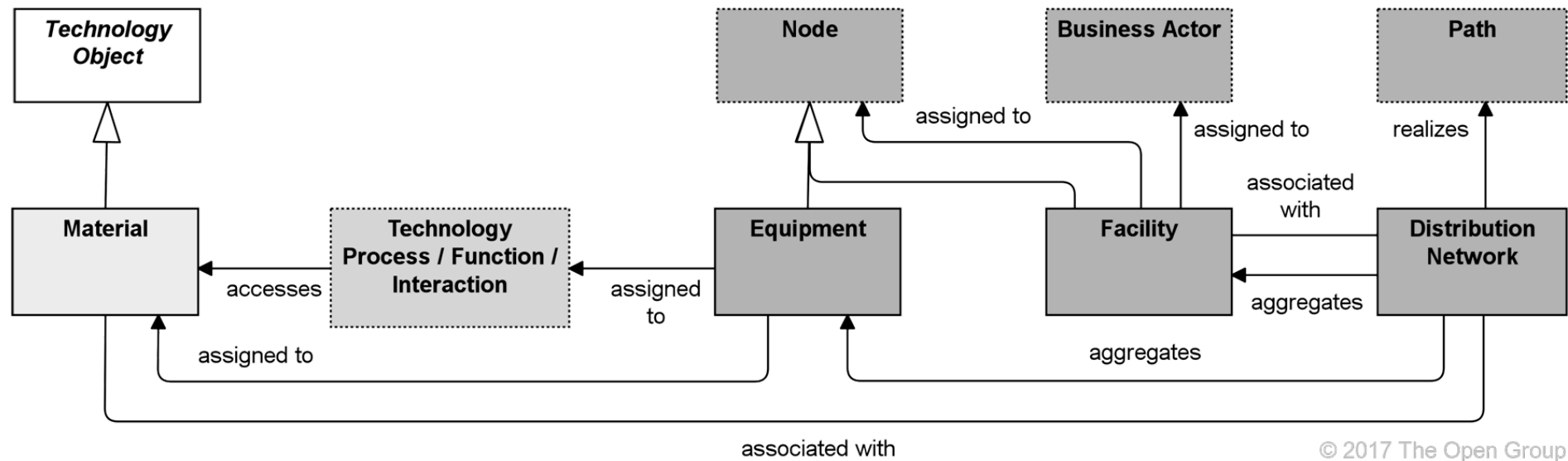
Physical elements

| Element	Definition	Notation
Equipment	One or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.	
Facility	A physical structure or environment.	
Distribution network	A physical network used to transport materials or energy.	
Material	Tangible physical matter or physical elements.	

Example Physical elements
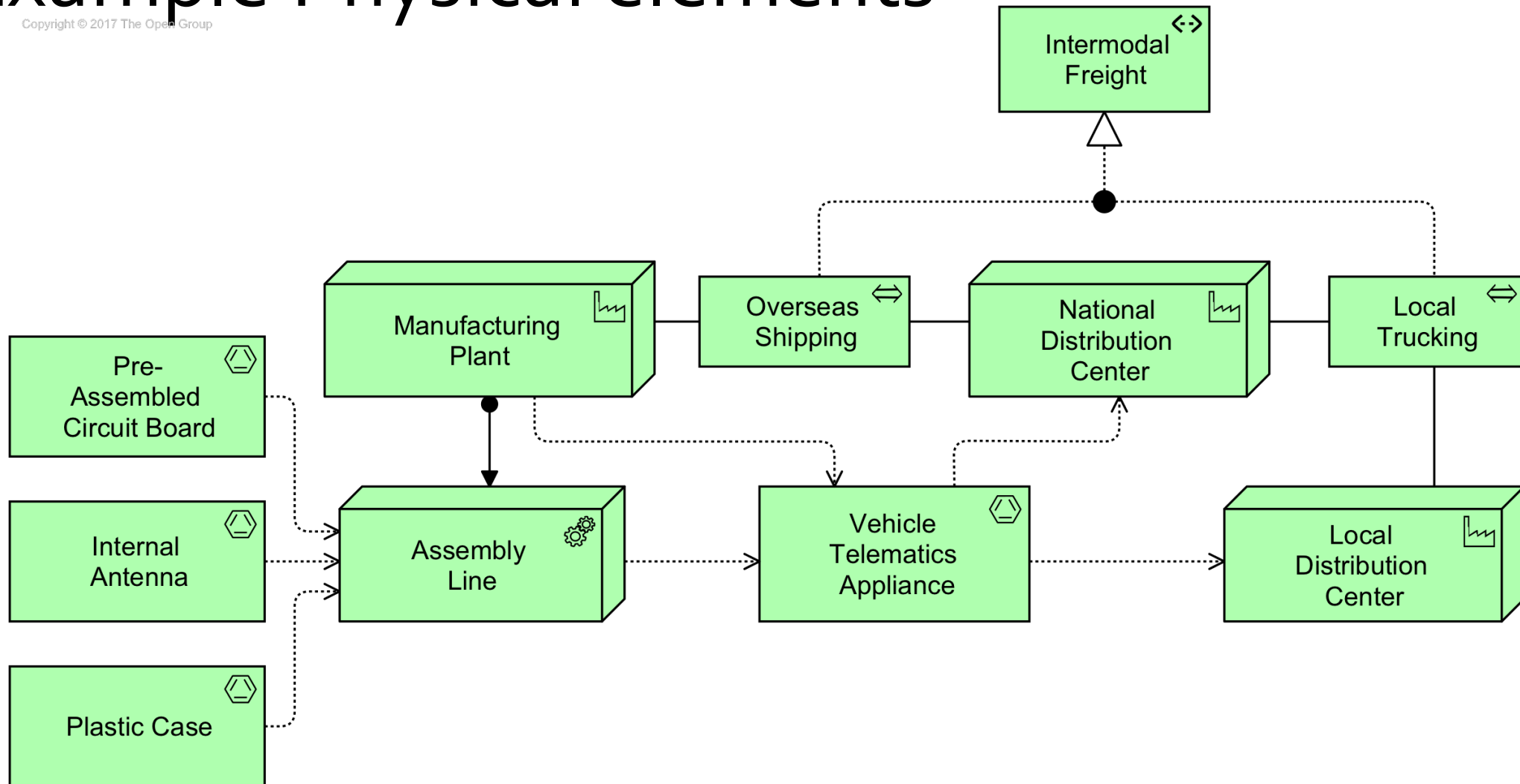
Cross-Layer Dependences
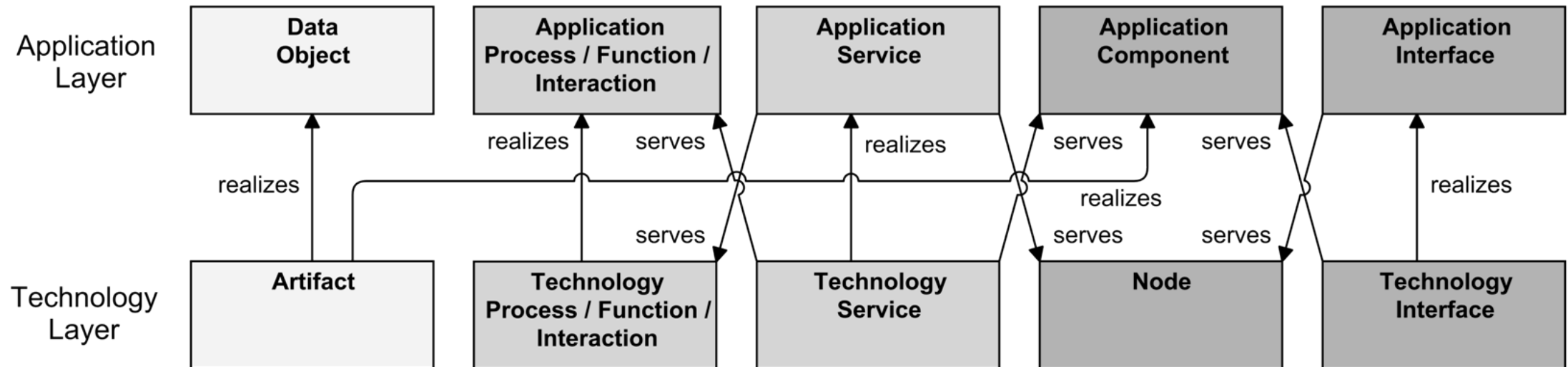
- Alignment of Business Layer and Lower Layers
- Alignment of Application and Technology Layers
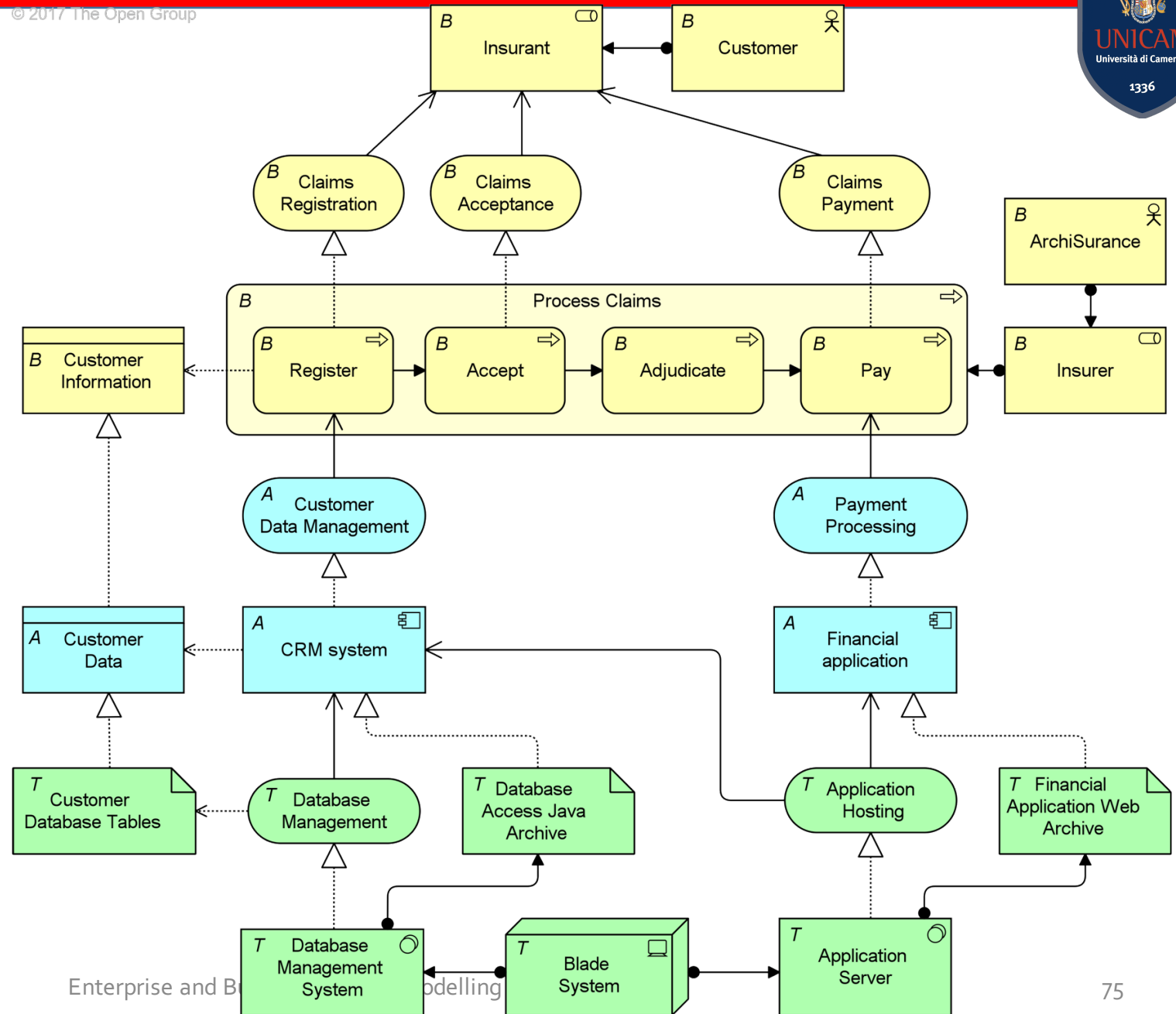
Alignment of Business Layer and Lower Layers
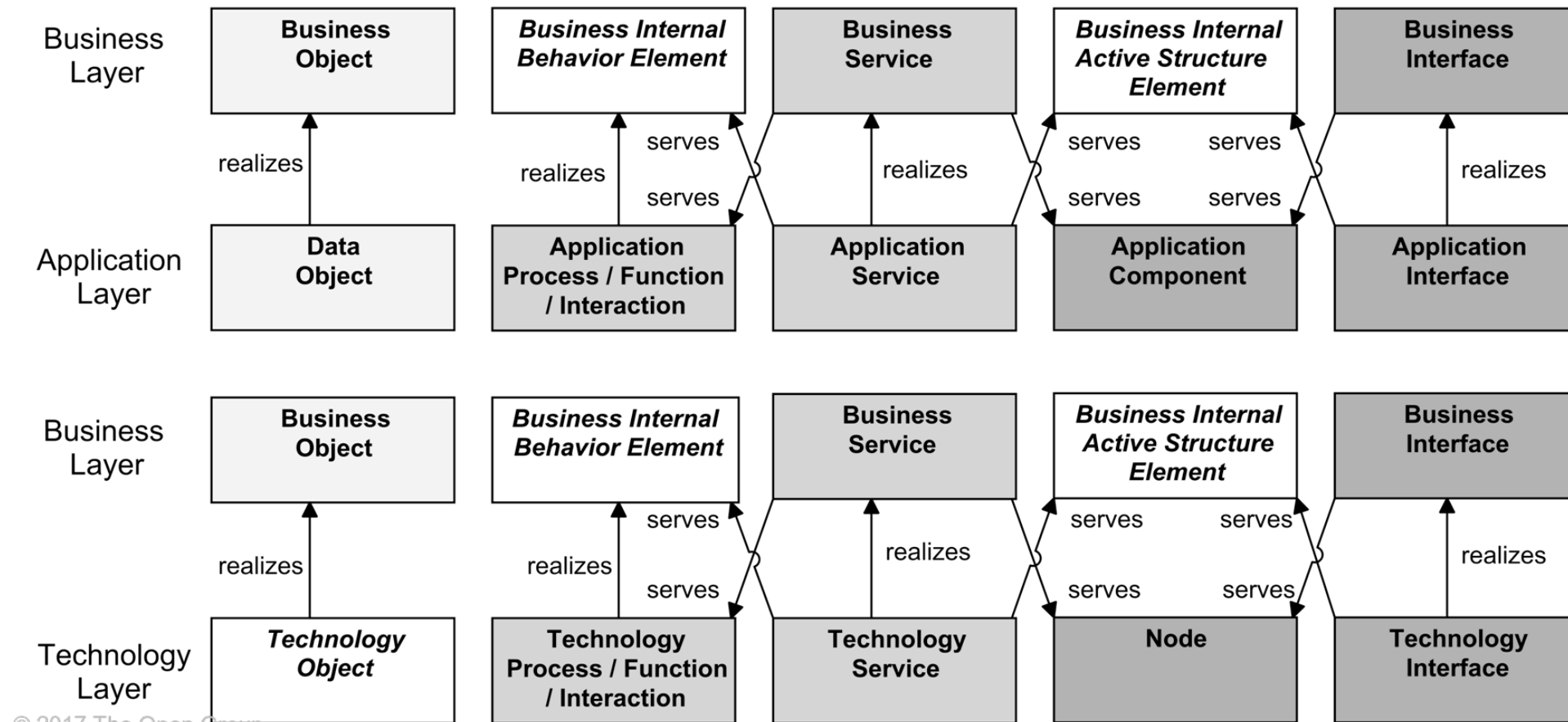
Alignment of Application and Technology Layers

© 2017 The Open Group
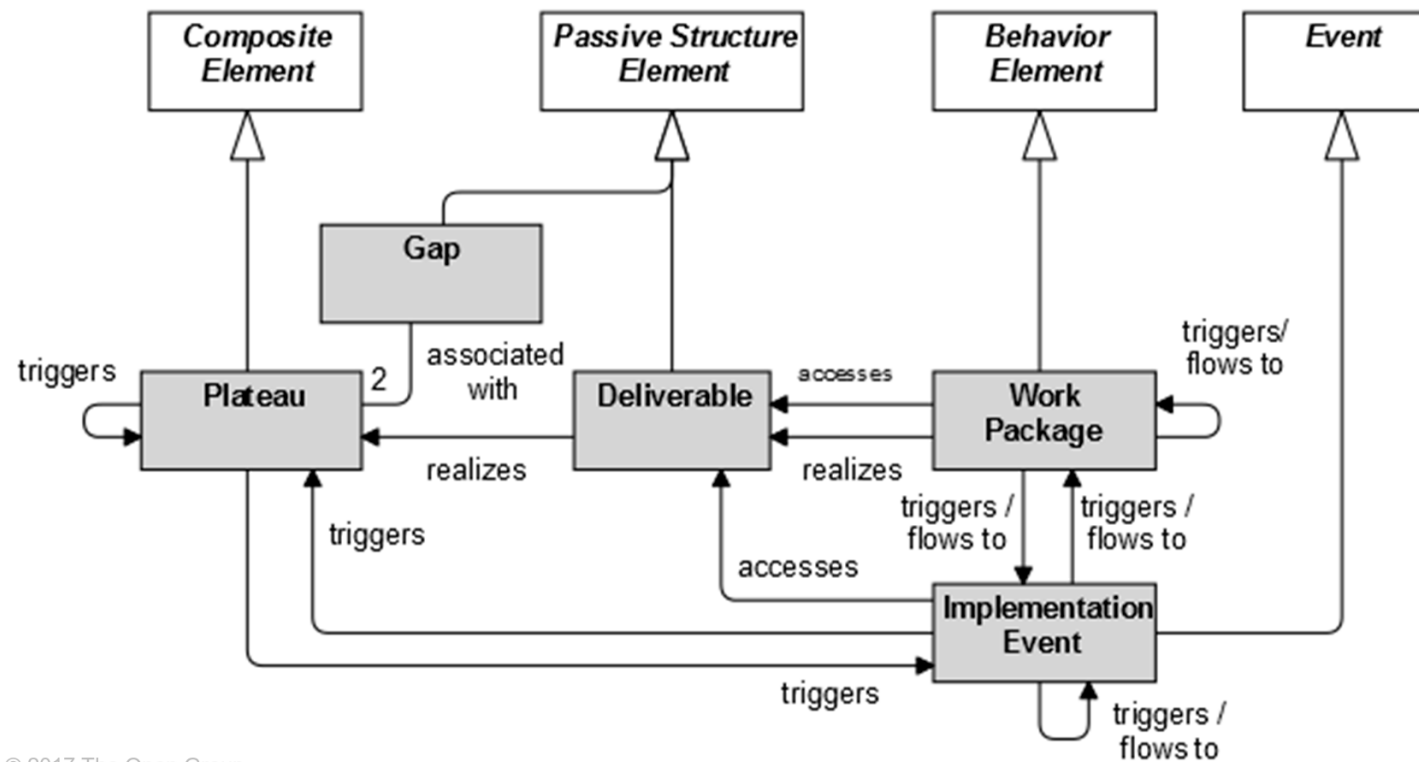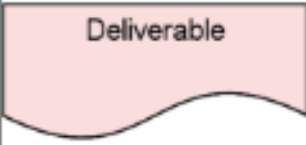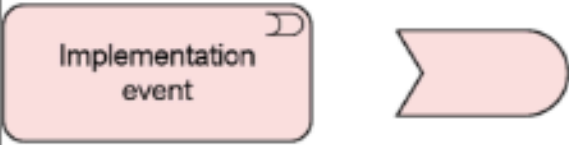
Examples of Cross – Layer Alignment
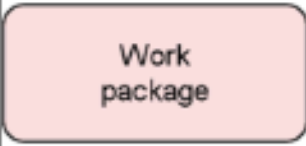
Implementation and Migration Meta-model

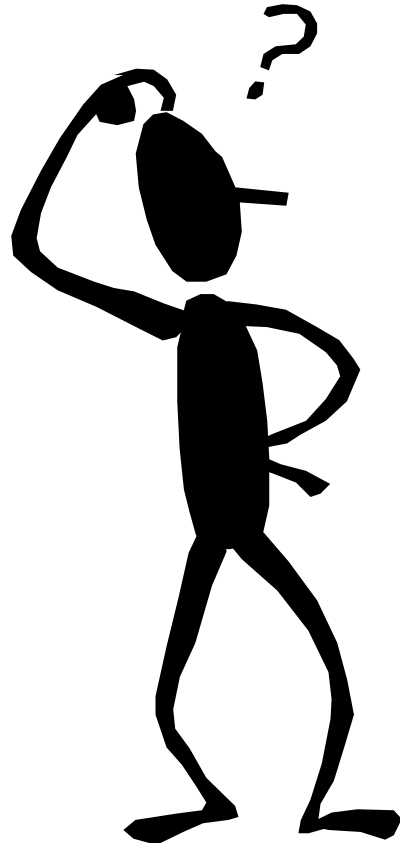© 2017 The Open Group

Implementation and Migration Elements

Element	Definition	Notation
Work package	A series of actions identified and designed to achieve specific results within specified time and resource constraints.	Work package
Deliverable	A precisely-defined outcome of a work package.	Deliverable
Implementation event	A behavior element that denotes a state change related to implementation or migration.	Implementation event
Plateau	A relatively stable state of the architecture that exists during a limited period of time.	Plateau
Gap	A statement of difference between two plateaus.	Gap

Example Implementation and Migration Elements

78

Questions?