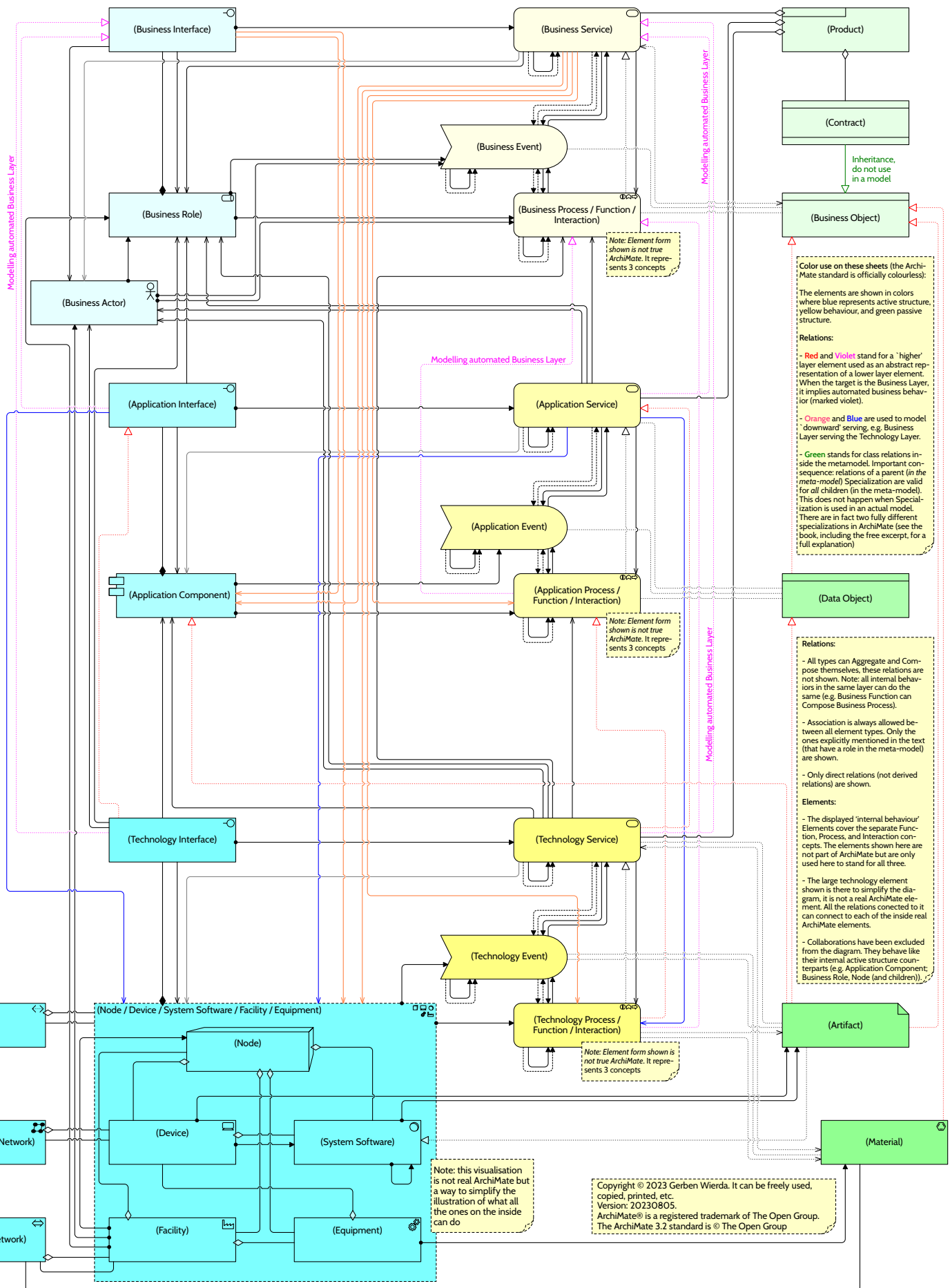# ArchiMate® 3.2 Metamodel – Core
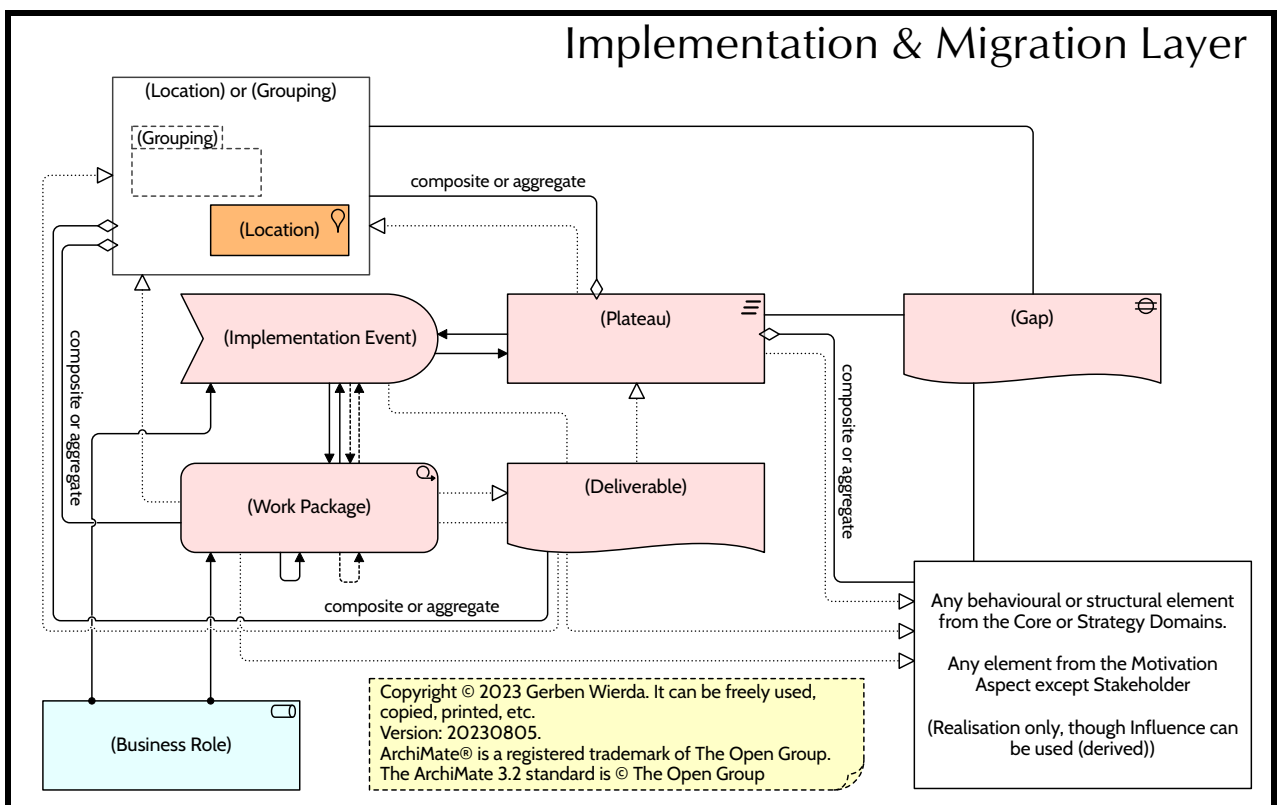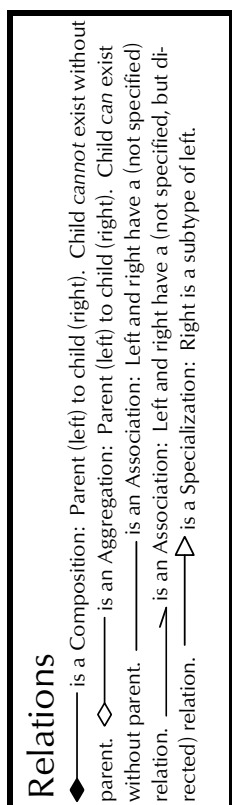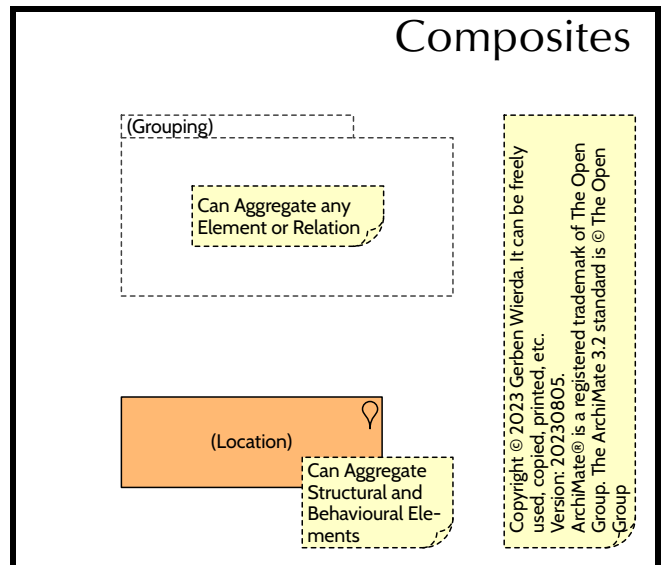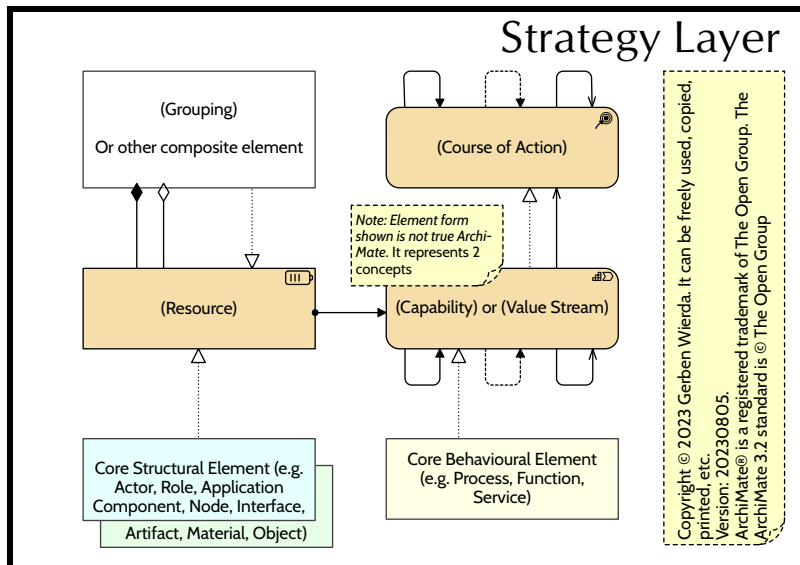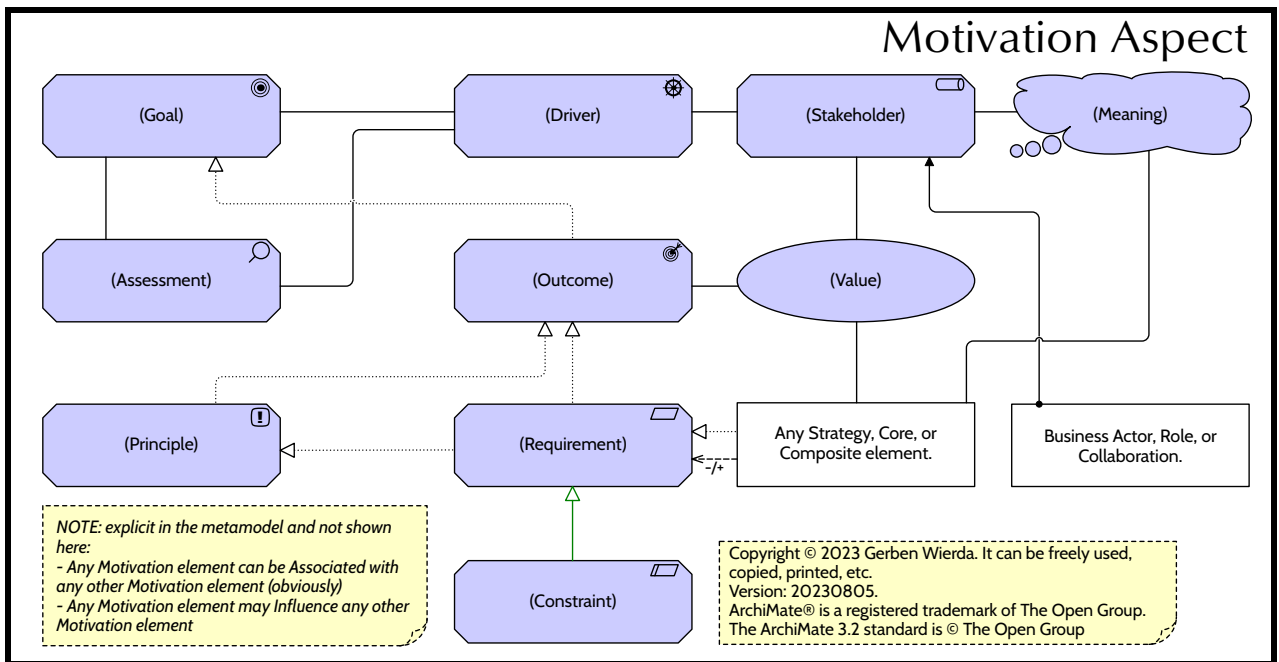
Only direct relations are shown. Abstract elements are expanded so only real metamodel elements are shown. Uses the **Mastering ArchiMate 9-colour scheme** to assist learning and quick reading.
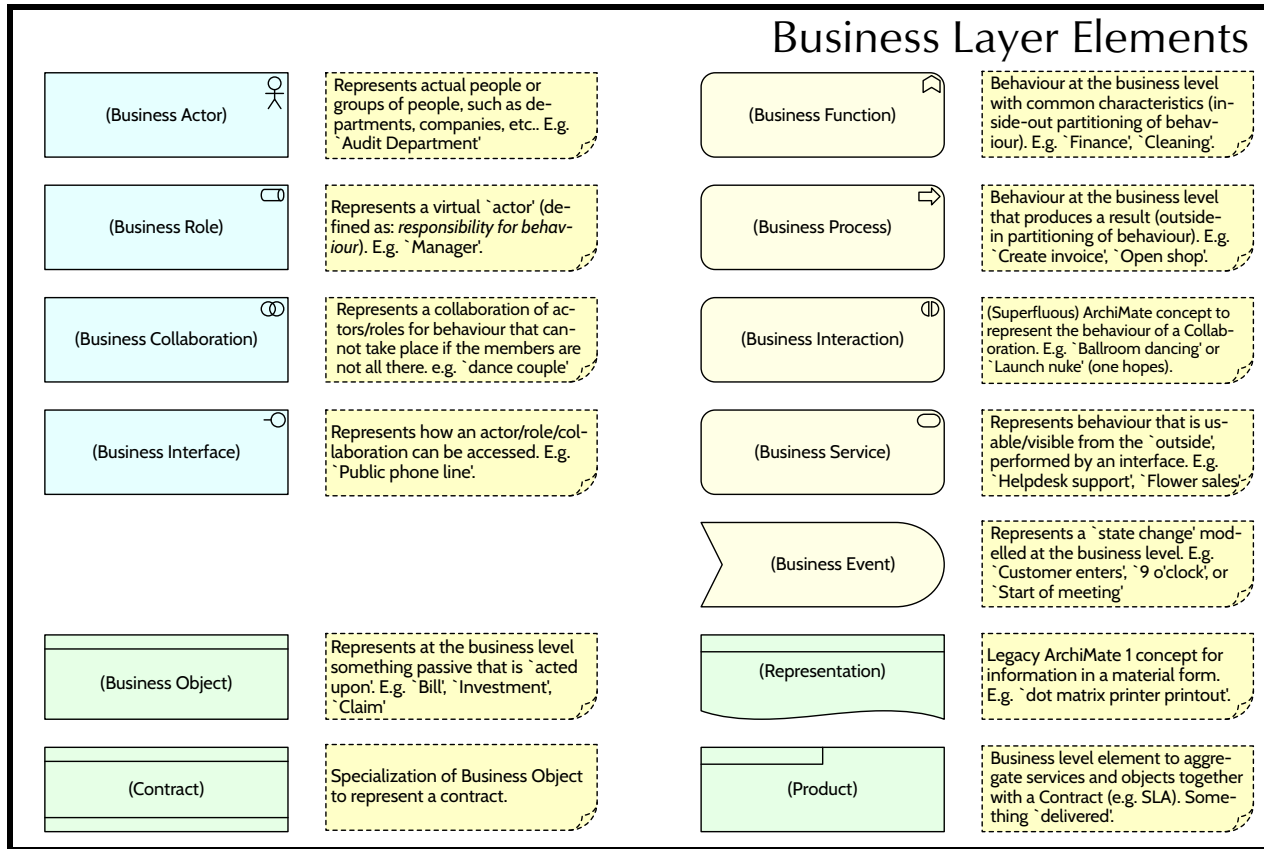
(Business Interface)

(Business Service)

(Product)

(Contract)

(Business Event)

(Business Role)

Inheritance, do not use in a model

(Business Process / Function / Interaction)

Note: Element form shown is not true ArchiMate. It represents 3 concepts

(Business Object)

(Business Actor)

Modelling automated Business Layer

Modelling automated Business Layer

Modelling automated Business Layer

Color use on these sheets (the Archi-Mate standard is officially colourless):

The elements are shown in colors where blue represents active structure, yellow behaviour, and green passive structure.

Relations:

- **Red** and **Violet** stand for a `higher' layer element used as an abstract representation of a lower layer element. When the target is the Business Layer, it implies automated business behavior (marked violet).

- **Orange** and **Blue** are used to model `downward' serving, e.g. Business Layer serving the Technology Layer.

- **Green** stands for class relations in-sequence: relations of a parent (*in the meta-model*) Specialization are valid for *all* children (in the meta-model). This does not happen when Specialization is used in an actual model. There are in fact two fully different specializations in ArchiMate (see the book, including the free excerpt, for a full explanation)

(Application Interface)

(Application Service)

(Application Event)

(Application Component)

(Application Process / Function / Interaction)

Note: Element form shown is not true ArchiMate. It represents 3 concepts

(Data Object)

Relations:

- All types can Aggregate and Compose themselves, these relations are not shown. Note: all internal behaviors in the same layer can do the same (e.g. Business Function can Compose Business Process).

- Association is always allowed between all element types. Only the ones explicitly mentioned in the text (that have a role in the meta-model) are shown.

- Only direct relations (not derived relations) are shown.

Elements:

- The displayed 'internal behaviour' Elements cover the separate Function, Process, and Interaction concepts. The elements shown here are not part of ArchiMate but are only used here to stand for all three.

- The large technology element shown is there to simplify the diagram, it is not a real ArchiMate element. All the relations connected to it can connect to each of the inside real ArchiMate elements.

- Collaborations have been excluded from the diagram. They behave like their internal active structure counterparts (e.g. Application Component; Business Role, Node (and children)).

(Technology Interface)

(Technology Service)

(Technology Event)

(Path)

(Node / Device / System Software / Facility / Equipment)

(Node)

(Technology Process / Function / Interaction)

Note: Element form shown is not true ArchiMate. It represents 3 concepts

(Artifact)

(Communication Network)

(Device)

(System Software)

(Distribution Network)

(Facility)

(Equipment)

(Material)

Note: this visualisation is not real ArchiMate but a way to simplify the illustration of what all the ones on the inside can do

Diagrams by **Gerben Wierda**.

This is the very fine print. Document locked against editing due to external licensing demands.

ArchiMate® 3.2 Metamodel – Non-Core
Diagrams by **Gerben Wierda**.

Motivation Aspect

(Goal)

(Driver)

(Stakeholder)

(Meaning)

(Assessment)

(Outcome)

(Value)

(Principle)

(Requirement)

Any Strategy, Core, or Composite element.

Business Actor, Role, or Collaboration.

-/+

(Constraint)

NOTE: explicit in the metamodel and not shown here:
- Any Motivation element can be Associated with any other Motivation element (obviously)
- Any Motivation element may Influence any other Motivation element

Strategy Layer

(Grouping)
Or other composite element

(Course of Action)

Note: Element form shown is not true Archi-Mate. It represents 2 concepts

(Resource)

(Capability) or (Value Stream)

Core Structural Element (e.g. Actor, Role, Application Component, Node, Interface, Artifact, Material, Object)

Core Behavioural Element (e.g. Process, Function, Service)

Composites

(Grouping)

Can Aggregate any Element or Relation

(Location)

Can Aggregate Structural and Behavioural Elements

Relations

— is a Composition: Parent (left) to child (right). Child *cannot* exist without parent.

◇ is an Aggregation: Parent (left) to child (right). Child *can* exist without parent.

— is an Association: Left and right have a (not specified) relation.

→ is an Association: Left and right have a (not specified, but directed) relation.

△ is a Specialization: Right is a subtype of left.

Implementation & Migration Layer

(Location) or (Grouping)

(Grouping)

(Location)

composite or aggregate

(Implementation Event)

(Plateau)

(Gap)

(Work Package)

(Deliverable)

composite or aggregate

composite or aggregate

composite or aggregate

(Business Role)

Any behavioural or structural element from the Core or Strategy Domains.

Any element from the Motivation Aspect except Stakeholder

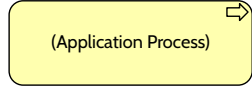(Realisation only, though Influence can be used (derived))

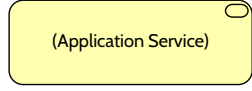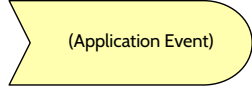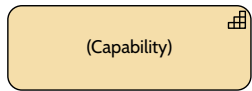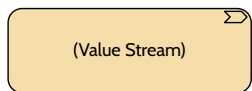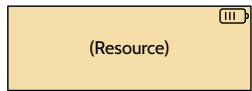# Overview of ArchiMate® 3.2 Elements (1/2)
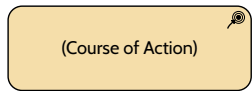Diagrams by **Gerben Wierda**.

Business Layer Elements

Element	Description
(Business Actor)	Represents actual people or groups of people, such as departments, companies, etc.. E.g. `Audit Department'
(Business Role)	Represents a virtual `actor' (defined as: *responsibility for behaviour*). E.g. `Manager'.
(Business Collaboration)	Represents a collaboration of actors/roles for behaviour that cannot take place if the members are not all there. e.g. `dance couple'
(Business Interface)	Represents how an actor/role/collaboration can be accessed. E.g. `Public phone line'.
(Business Object)	Represents at the business level something passive that is `acted upon'. E.g. `Bill', `Investment', `Claim'
(Contract)	Specialization of Business Object to represent a contract.
(Business Function)	Behaviour at the business level with common characteristics (inside-out partitioning of behaviour). E.g. `Finance', `Cleaning'.
(Business Process)	Behaviour at the business level that produces a result (outside-in partitioning of behaviour). E.g. `Create invoice', `Open shop'.
(Business Interaction)	(Superfluous) ArchiMate concept to represent the behaviour of a Collaboration. E.g. `Ballroom dancing' or `Launch nuke' (one hopes).
(Business Service)	Represents behaviour that is usable/visible from the `outside', performed by an interface. E.g. `Helpdesk support', `Flower sales'
(Business Event)	Represents a `state change' modelled at the business level. E.g. `Customer enters', `9 o'clock', or `Start of meeting'
(Representation)	Legacy ArchiMate 1 concept for information in a material form. E.g. `dot matrix printer printout'.
(Product)	Business level element to aggregate services and objects together with a Contract (e.g. SLA). Something `delivered'.

Relations

is an Assignment: Left is responsible to right, or right exists at left. ───── is Access: Direction always to `object', regardless of arrowheads. ╌╌╌╌> is read and write, ─────> is write when arrowhead at passive element and read otherwise. ──────▲ is a Trigger: Left triggers right. ┄┄┄┄▶ is a Flow: Something flows from left to right. A payload may be connected to (via Association) the relation. ╌╌╌+/-╌╌> is an Influence: Left influences right.

Technology Layer Elements

Element	Description
(Node)	Generic technology element. Often used to aggregate other technology elements, e.g. a server as Node that contains Device and System Software
(Device)	Represents actual computing hardware. E.g. `x86 blade'
(System Software)	Software system at the technology level. Generally used for platforms. E.g. `PostgreSQL', `Windows 2016 Server', `Java SE'
(Technology Interface)	How an active technology element can be accessed. E.g. `Port 25', `SMTP Protocol'
(Technology Collaboration)	(Unnecessary) concept that represents a combination of technology elements that only together can perform certain behaviour.
(Path)	Logical representation of a distribution channel between Nodes etc.. E.g. `data replication', `finished product exchange'
(Communication Network)	Data transport facility. E.g. `datacenter network'
(Facility)	Higher level grouping of physical actors. E.g. `Brick factory'.
(Equipment)	Machinery that can perform physical behaviour (as opposed to informational behaviour). E.g. `Steel mill'.
(Technology Function)	Behaviour at the technology level with common characteristics (inside-out partitioning of behaviour). E.g. `DB backup', `Machine sawing'.
(Technology Process)	Behaviour at the technology level that produces a result (outside-in partitioning of behaviour). E.g. `Delete database', `Build closet'.
(Technology Service)	Represents `accessible' (by persons or IT) behaviour of `technology'. Performed by an interface. E.g. `Use RDBMS', `Accept waste'
(Technology Event)	Represents a `state change' modelled at the technology level. E.g. `every 10 minutes', `server boots', `tank empty'
(Technology Interaction)	(Unnecessary) concept that represents the behaviour of a Technology Collaboration.
(Artifact)	Raw data. E.g. `database files', `software distribution', `PDF file'
(Material)	Physical stuff. E.g. `Wood blocks', `Nails'
(Distribution Network)	Means of physical distribution. E.g. `rail', `conveyor belt'

NOTE: The element descriptions on this page are not the official ones from The Open Group and may be 'opinionated'.

Overview of ArchiMate® 3.2 Elements (2/2)

Diagrams by **Gerben Wierda**.

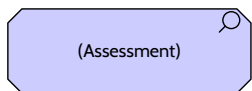## Application Layer Elements
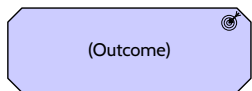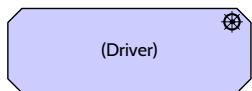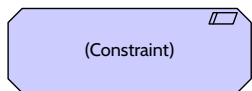
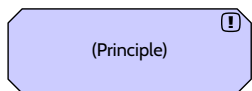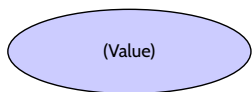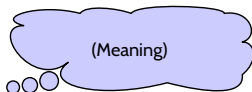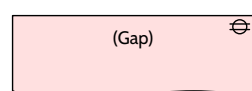| Element	Description
(Application Component)	Represents a software system at the application level. E.g. `EA Modelling System' or `BiZZdesign Enterprise Studio'
(Application Interface)	Represents the interface of an application: how `actors' (both persons and IT) can interact with the system. E.g. `GUI', `API'
(Application Collaboration)	(Unnecessary) concept that represents a combination of applications that only together can perform certain behaviour.
(Data Object)	Element that represents information at the application level. E.g. `customer record', `transaction record'
(Application Function)	Behaviour at the application level with common characteristics (inside-out partitioning of behaviour). E.g. `risk calculations'.
(Application Process)	Behaviour at the application level that produces a result (outside-in partitioning of behaviour). E.g. `calculate monthly payment'.
(Application Service)	Represents `accessible' (by persons or IT) behaviour of an application. Performed by an interface. E.g. `Website', `RESTful customer handler'
(Application Event)	Represents a `state change' modelled at the application level. E.g. `every 10 minutes', `application starts', `transaction received'
(Application Interaction)	(Unnecessary) concept that represents the behaviour of an Application Collaboration.
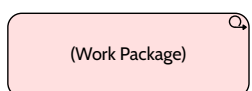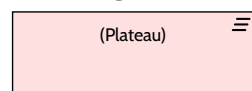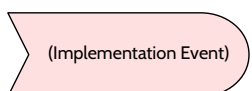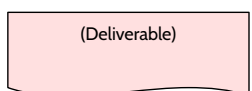
Strategy Layer Elements

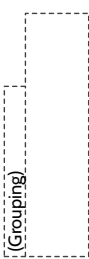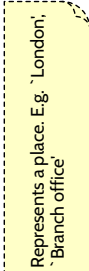| Element	Description
(Capability)	Represents a high-level ability an organisation has. E.g. `build roads'
(Resource)	Strategic representation of an asset of the organisation. E.g. `Wind mill field'
(Value Stream)	Represents how an organisation is set up to provide value to its stakeholders (generally customers)
(Course of Action)	Documentation/description of something strategic/tactical the organisation has decided to do. E.g. `acquire complementary company'

Motivation Aspect Elements

Element	Description
(Goal)	Desired outcome. E.g. `10% increase in ROI year X versus year Y'
(Outcome)	Actual Outcome of what an organisation does. E.g. `10% increase in ROI year X versus year Y'
(Driver)	Represents a reason for change and for (architectural) decisions.
(Assessment)	Formally: Represents how we stand in relation to a certain Driver. But why only Driver?
(Value)	Describes/documents the value (monetary or otherwise) of another concept.
(Stakeholder)	Superfluous concept (could have been a Role) that represents someone with a stake in the architecture. E.g. `Product Owner'
(Requirement)	Represents requirements that have to be fulfilled by the architecture. E.g. `Use supported software'
(Constraint)	Superfluous (is just another Requirement) negative Requirement. E.g. `Do not use unsupported software'
(Principle)	Very strong Requirement. Be careful with these, can be toxic. E.g. `never use principles'
(Meaning)	Philosophically suspect concept from ArchiMate 1. Formally: represents the intent of something.

Implementation & Migration Layer Elements

Element	Description
(Work Package)	Represent an amount of work with a clear start and end date and (like Process) a clear outcome.
(Implementation Event)	Represents a `state change' in business transformation. E.g. `Release to production'
(Deliverable)	The result of a Work Package
(Plateau)	Represents a `state' of the landscape that exists during a certain amount of time.
(Gap)	Represents the difference between two (successive) Plateaus

Composite Elements

Element	Description
Can Aggregate or Compose anything	
(Grouping)	
(Location)	Represents a place. E.g. `London', `Branch office'

NOTE: The element descriptions on this page are not the official ones from The Open Group and may be 'opinionated'.