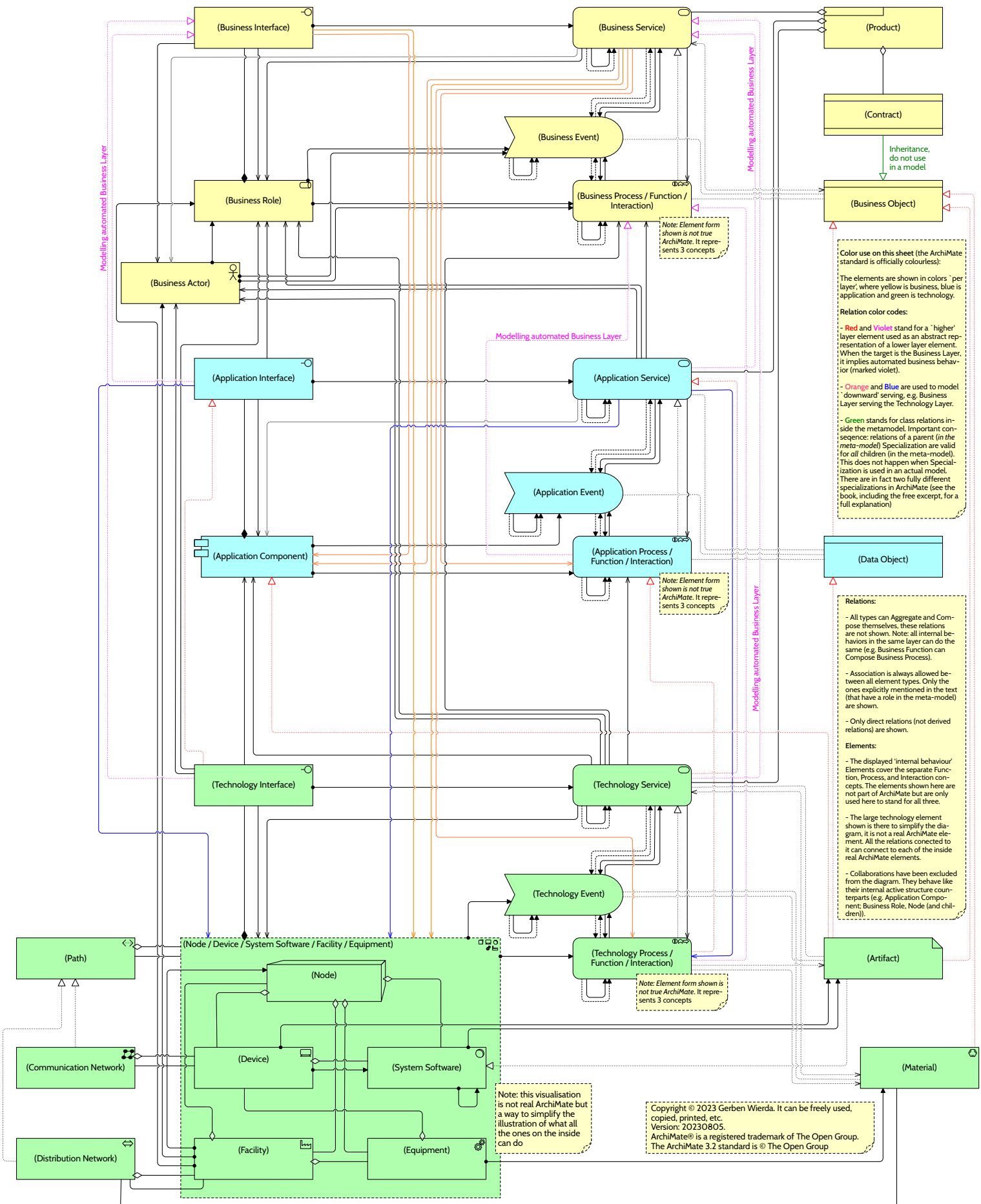


ArchiMate® 3.2 Metamodel – Core

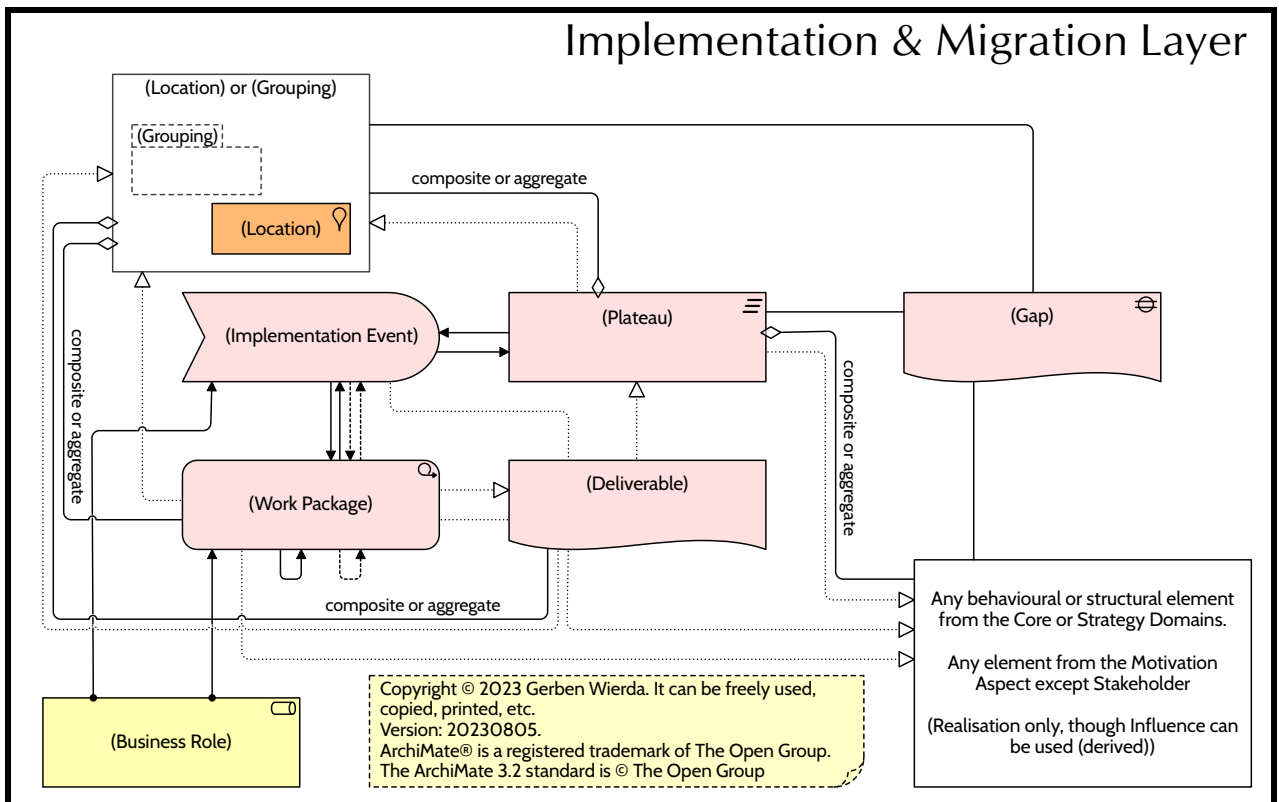
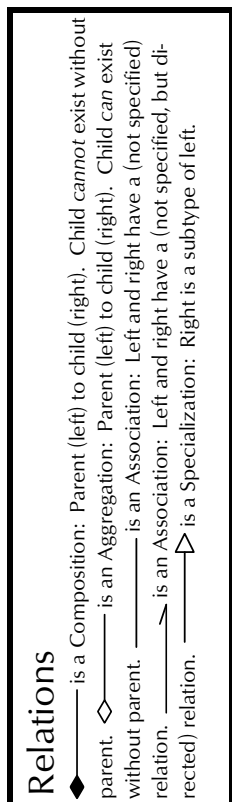
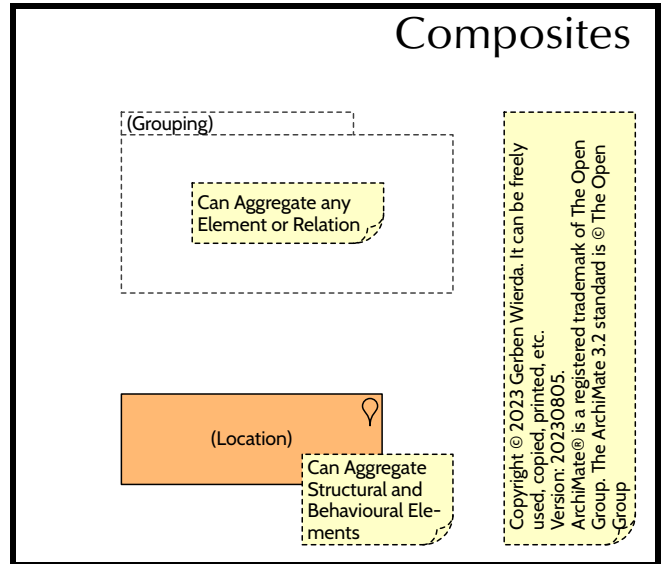
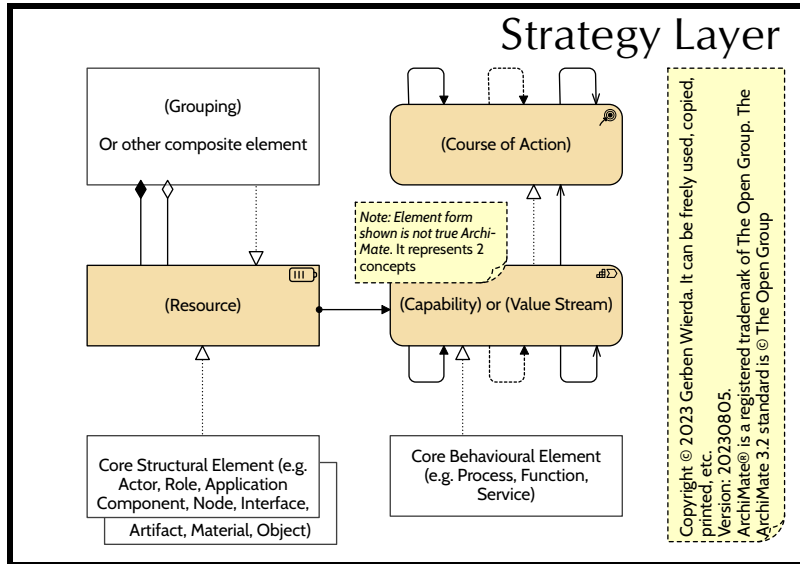
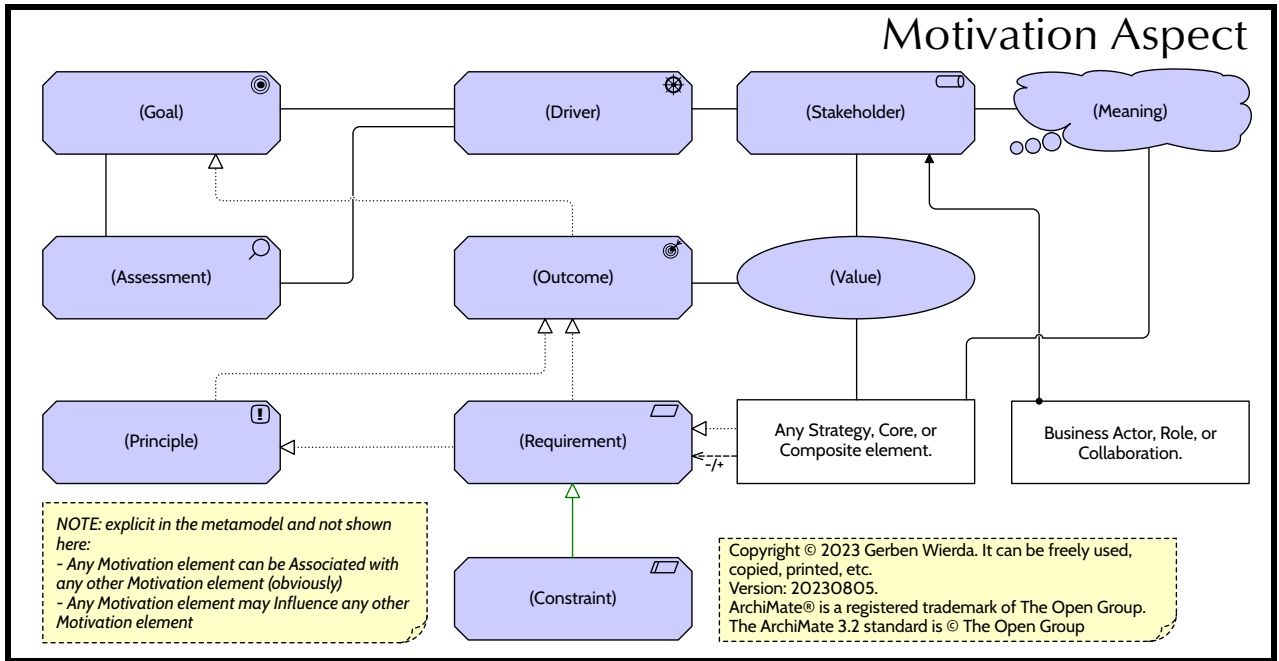
Only direct relations are shown. Abstract elements are expanded so only real metamodel elements are shown. The elements are shown in colors 'per layer', where yellow is business, blue is application and green is technology.



ArchiMate® 3.2 Metamodel – Non-Core

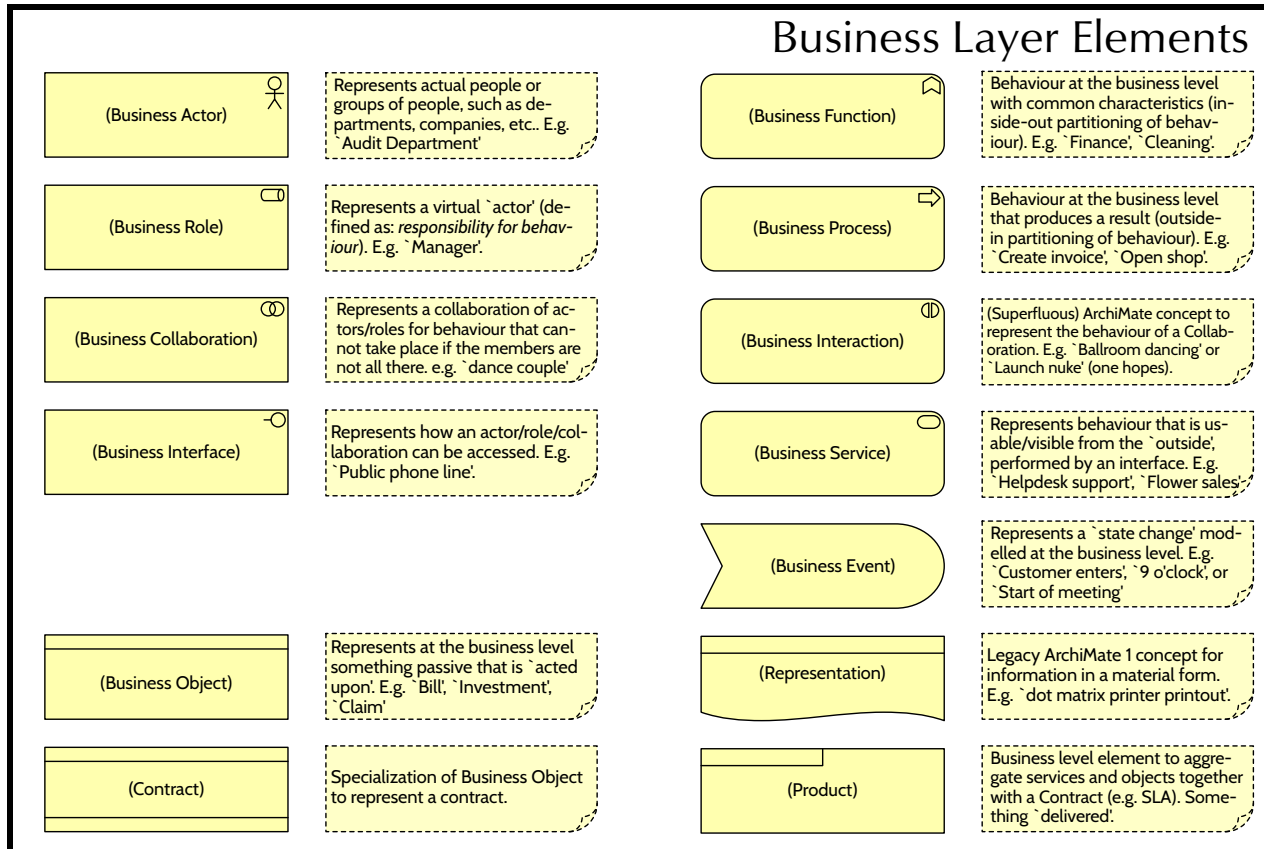
Diagrams by Gerben Wierda.

Copyright © 2023 Gerben Wierda. It can be freely used, copied, printed, etc.
 Version: 20230805-en.
 ArchiMate® is a registered trademark of The Open Group. The ArchiMate 3.2 standard is © The Open Group

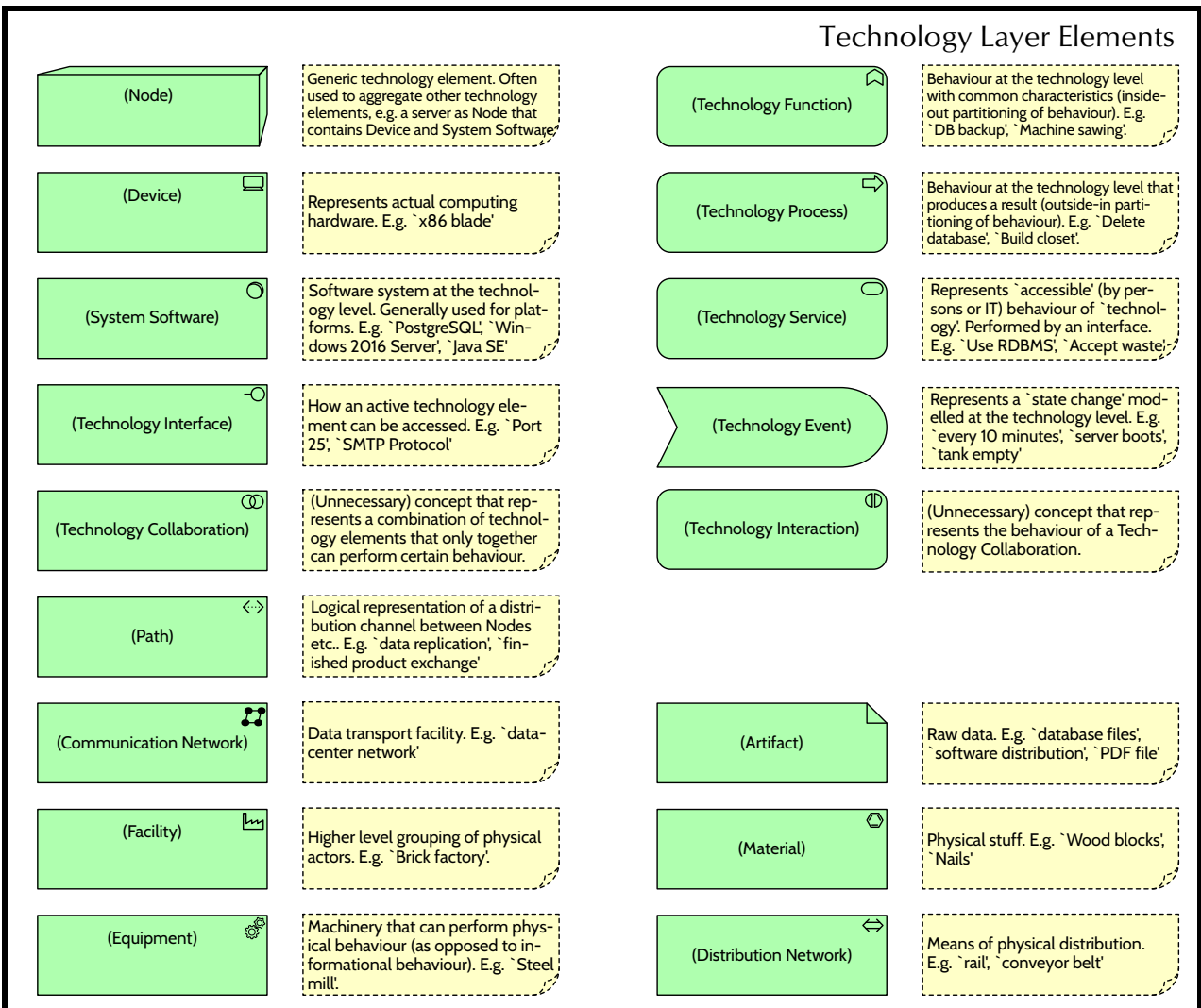
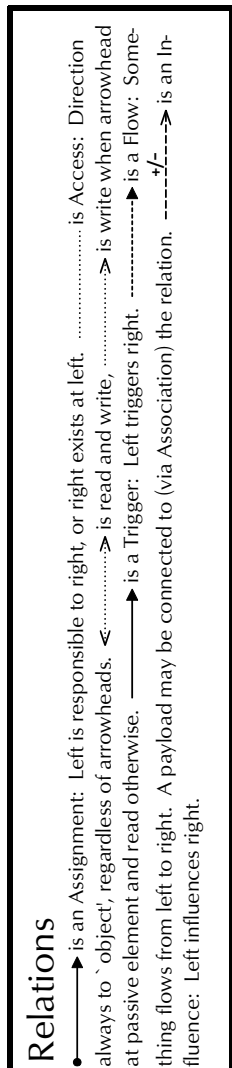


Overview of ArchiMate® 3.2 Elements (1/2)

Diagrams by **Gerben Wierda**.



Copyright © 2023 Gerben Wierda. It can be freely used, copied, printed, etc.
 Version: 20230805-en.
 ArchiMate® is a registered trademark of The Open Group. The ArchiMate 3.2 standard is © The Open Group





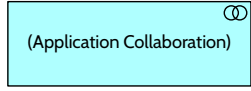
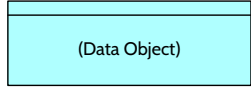
NOTE: The element descriptions on this page are not the official ones from The Open Group and may be 'opinionated'.



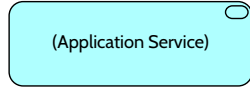
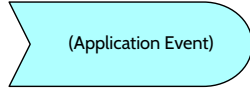

Overview of ArchiMate® 3.2 Elements (2/2)

Diagrams by **Gerben Wierda**.

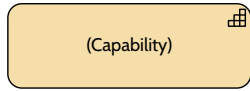
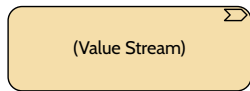
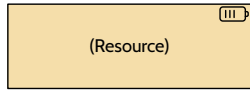
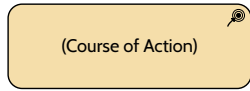
Copyright © 2023 Gerben Wierda. It can be freely used, copied, printed, etc.
Version: 20230805-en.
ArchiMate® is a registered trademark of The Open Group.
The ArchiMate 3.2 standard is © The Open Group

Application Layer Elements



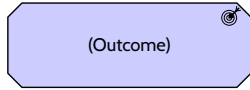


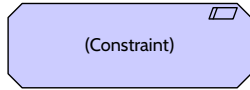
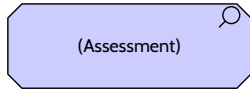
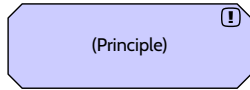
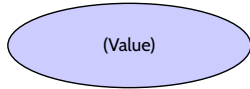
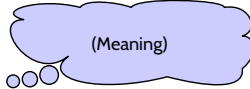
 (Application Component)	Represents a software system at the application level. E.g. 'EA Modelling System' or 'BIZZdesign Enterprise Studio'
 (Application Interface)	Represents the interface of an application: how 'actors' (both persons and IT) can interact with the system. E.g. 'GUI', 'API'
 (Application 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.

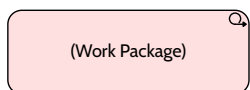
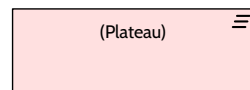
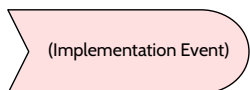
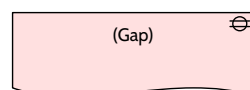
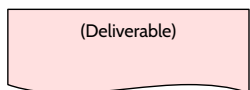
Strategy Layer Elements

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


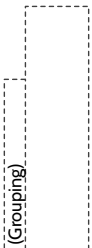
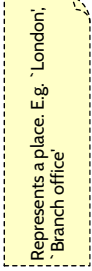
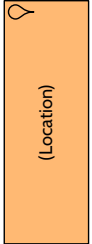
Motivation Aspect Elements

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

Implementation & Migration Layer Elements

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

Composite Elements

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

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