

# ArchiMate cheat sheet — every element, by layer and aspect

All 60 elements of ArchiMate® 3.2 in one grid, placed by the two questions that decide where an element belongs.

## THE FRAMEWORK

layer answers **whom it serves** · aspect answers **what kind of thing it is**

|                                                                 | ACTIVE STRUCTURE<br>who or what does it                                                                          | BEHAVIOUR<br>what is done                                                                                      | PASSIVE STRUCTURE<br>what it is done to                        |
|-----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Motivation</b><br>why — its own aspect                       | Stakeholder · Driver · Assessment · Goal · Outcome · Principle · Requirement · Constraint · Meaning · Value      |                                                                                                                |                                                                |
| <b>Strategy</b><br>what we must be able to do                   | Resource                                                                                                         | Capability · Value Stream · Course of Action                                                                   | —                                                              |
| <b>Business</b><br>what the organisation does for its customers | Business Actor · Business Role · Business Collaboration · Business Interface                                     | Business Process · Business Function · Business Interaction · Business Event · Business Service                | Business Object · Contract · Representation · <b>Product</b> † |
| <b>Application</b><br>what software does for the business       | Application Component · Application Collaboration · Application Interface                                        | Application Function · Application Interaction · Application Process · Application Event · Application Service | Data Object                                                    |
| <b>Technology</b><br>what infrastructure does for software      | Node · Device · System Software · Technology Collaboration · Technology Interface · Path · Communication Network | Technology Function · Technology Process · Technology Interaction · Technology Event · Technology Service      | Artifact                                                       |
| <b>Physical</b><br>part of Technology, not a layer of its own   | Equipment · Facility · Distribution Network                                                                      | — <i>uses Technology behaviour</i>                                                                             | Material                                                       |
| <b>Implementation &amp; Migration</b><br>how we get there       | —                                                                                                                | Work Package · Implementation Event                                                                            | Deliverable · <b>Plateau</b> ‡ · <b>Gap</b> ‡                  |
| <b>Composite</b><br>no layer of their own                       | Grouping — anything that belongs together · Location — where elements sit or behaviour happens                   |                                                                                                                |                                                                |

† **Product** is **composite**, not passive structure: it aggregates services and passive elements across layers, under a contract. Most cheat sheets file it wrongly. ‡ **Plateau** aggregates core elements as they stand at one point in time; **Gap** states the difference between two plateaus.

## PLACING AN ELEMENT — TWO QUESTIONS

**1. Whom does it serve?** That is the layer. Technology serves applications, applications serve the business, the business serves its customers. **2. What kind of thing is it?** Active structure does something, behaviour is what is done, passive structure is what it is done to.

## INTERNAL OR EXTERNAL — THE PAIR PEOPLE MISS

Every layer repeats the same pair. A **Service** is the outward face of **behaviour** — what others may use. An **Interface** is the outward face of **active structure** — where they reach it. Everything else in the layer is internal, and no one outside should depend on it.

## READING THE NOTATION

**Colour is convention, not specification** — yellow business, blue application, green technology, purple motivation, tan strategy, pink implementation. The small **icon in the top-right** corner of a box carries the actual element type; a rounded box is behaviour, a square one is structure.

## RELATIONSHIPS

11 + junction

### STRUCTURAL — STATIC COHERENCE

-  **Composition**  
the part cannot exist without the whole
-  **Aggregation**  
groups parts that can exist alone
-  **Assignment**  
who performs it, where it runs
-  **Realization**  
the concrete one delivers the abstract one

### DEPENDENCY — ONE NEEDS ANOTHER

-  **Serving**  
provides its functionality to another
-  **Access**  
reads, writes or deletes a passive element
-  **Influence**  
affects a motivation element, for or against
-  **Association**  
a link the language has no word for

### DYNAMIC — SOMETHING MOVES

-  **Triggering**  
this then that — order in time
-  **Flow**  
something is transferred — data, money, goods

### OTHER & CONNECTOR

-  **Specialization**  
is a particular kind of
-  **Junction**  
and / or — joins relationships of one type

### The derivation rule — the one worth memorising

A chain of relationships pointing the same way can be replaced by the **weakest link in the chain**. That is how a diagram stays readable when you zoom out.  
composition > aggregation > assignment > realization > serving > access > influence > association

### THREE PAIRS THAT GET CONFUSED

**Assignment vs realization.** Assignment allocates a performer — an actor *does* a process. Realization says something concrete delivers something abstract — a process *provides* a service.  
**Serving vs flow.** Serving is a standing offer: it is available to you. Flow is one actual transfer during one run.  
**Composition vs aggregation.** A composed part dies with its whole and has exactly one whole. An aggregated part survives on its own and may belong to several.