

10.2 Traceability Across Layers

[Go to Top](#)

[Traceability](#) across layers preserves architectural meaning as later parts specialise, realise, deploy, and provide [Evidence](#) for the conceptual architecture.

Each layer has a distinct [Traceability](#) responsibility:

Table 10.2-1: Traceability responsibilities across architectural layers

Layer	Traceability responsibility
Conceptual Architecture	Defines stable concepts and conceptual boundaries.
Logical Architecture / Platform Independent Model (PIM)	Specialises conceptual elements into platform-independent logical structures, relationships, and constraints.
Implementation Profile / Platform Specific Model (PSM)	Maps logical elements to selected implementation technologies and artefacts.
Deployment Profile	Describes how implementation artefacts run in a concrete environment.
Evidence	Supports claims about architecture, implementation, deployment, execution, Interpretation , Governance and Authority , or review.

[Traceability](#) keeps these layers connected without allowing one layer to redefine another. A logical element may specialise a conceptual element. An implementation artefact may realise a logical element. A deployment artefact may execute or configure an implementation artefact. [Evidence](#) may support a claim about any of these layers.

From:
<https://wiki.didosolutions.com/> - Dido Solutions, Inc Wiki

Permanent link:
<https://wiki.didosolutions.com/fxdemo/01-part/10-conceptual-traceability-model/10-2-traceability-across-layers?rev=1782864251>

Last update: 2026/06/30 17:04

