

5. Reading the Architecture as an Onion

The architecture uses an onion structure. At the centre are durable concepts. The next layer defines logical relationships. The next layer specialises those relationships for a financial domain or demonstration. The next layer maps the logical model to selected technologies. The outer layer defines physical deployment, testability, and evidence. Figure 5-1 represents these layers.

Conceptual Architecture

- └ Logical Architecture / PIM
 - └ Domain Logical Profile
 - └ Implementation Profile / PSM
 - └ Deployment, Testability, and Evidence Plan

Figure 5-1: Layered structure of the Financial Systems Archetype document layers Each layer depends on the layer inside it. The outer layers may specialise, realise, deploy, or test inner-layer concepts. They must not redefine those concepts. Table 5-1: Examples of inner-layer concepts and later-layer realisations

Inner-layer concept	Later-layer realisation
Communication Endpoint	DDS Topic, API endpoint, message queue, event stream
Data Structure Definition	IDL structure, JSON Schema, XML Schema, UML/SysML model
Node	Process, container, pod, service, virtual machine
Runtime Plane	Topic namespace, DDS partition, Kubernetes namespace, network segment, security zone
Evidence	Logs, captured samples, test reports, deployment records, and Crucible records

The later-layer realisation does not replace the inner-layer concept.

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

Permanent link:
<https://wiki.didosolutions.com/fxdemo/00-part/06-reading-the-architecture-as-an-onion/start?rev=1782339471>

Last update: 2026/06/24 15:17

