

Part 1: Conceptual Architecture

[Go to The FX Demo Root](#)



Figure 1: FX Demo icon.

* [Foreword](#)

- [Introduction](#)
- [1. Scope](#)
- [2. Relationship to Part 0](#)
- [3. Traceability to Parent and Source Architectures](#)
 - [3.1 Traceability to SIP-RA](#)
 - [3.2 Alignment with FDIS-RA](#)
 - [3.3 Coverage of the Original FX Demo Reference Architecture](#)
 - [3.4 Traceability Position](#)
- 5. [4. Conceptual Architecture Principles](#)
 - [4.1 Separation of Concerns](#)
 - [4.2 Platform Independence](#)
 - [4.3 Classification Before Implementation](#)
 - [4.4 Logical Meaning Before Physical Realisation](#)
 - [4.5 Traceability](#)
 - [4.6 Evidence Orientation](#)
 - [4.7 Governance of Meaning](#)
- 6. [5. Archetype Classification Model](#)
 - [5.1 Overview](#)
 - [5.2 Ecosphere](#)
 - [5.3 Ecosystem](#)
 - [5.4 Domain](#)
 - [5.5 Classification Path](#)
 - [5.6 Classification Values and Runtime Identifiers](#)
 - [5.7 Classification and Governance](#)
- 7. [6. Financial Systems Classification](#)
 - [6.1 Overview](#)
 - [6.2 Finance Ecosphere](#)
 - [6.3 Monetary Ecosystem](#)

- 6.4 Foreign Exchange Domain
- 6.5 Classification Inheritance for FX Profiles
- 6.6 Future Financial Classifications
- 6.7 Classification Governance
- 8. 7. Core Conceptual Elements
 - 7.1 Overview
 - 7.2 Node
 - 7.3 Node Identity
 - 7.4 Node Role
 - 7.5 Runtime Plane
 - 7.6 Communication Endpoint
 - 7.7 Data Structure Definition
 - 7.8 Data Structure Instance
 - 7.9 Evidence
 - 7.10 Traceability
- 9. 8. Conceptual Runtime Plane Model
 - 8.1 Overview
 - 8.2 Control Plane
 - 8.3 Data Plane
 - 8.4 Health and Observability Plane
 - 8.5 Policy and Release Plane
 - 8.6 Audit and Provenance Plane
 - 8.7 Cross-Plane Relationships
- 10. 9. Separation of Concerns Model
 - 9.1 Overview
 - 9.2 Data
 - 9.3 Structure
 - 9.4 Semantics
 - 9.5 Interpretation
 - 9.6 Governance and Authority
 - 9.7 Traceability and Evidence
 - 9.8 Cross-Concern Relationships
- 11. 10. Conceptual Traceability Model
 - 10.1 Overview
 - 10.2 Traceability Across Layers
 - 10.3 Conceptual Element to Logical Element / PIM
 - 10.4 Logical Element / PIM to Implementation Artefact / PSM
 - 10.5 Implementation Artefact / PSM to Deployment Artefact
 - 10.6 Deployment Artefact to Evidence
 - 10.7 Bidirectional Review and Impact Analysis
 - 10.8 Traceability Boundaries
- 12. 11. Conceptual Rules and Constraints
 - 11.1 Overview
 - 11.2 Conceptual Elements Remain Distinct
 - 11.3 Conceptual Architecture Does Not Prescribe Realisation Mechanisms
 - 11.4 Classification Values Do Not Define Runtime Boundaries
 - 11.5 Runtime Planes Remain Conceptual Classifications
 - 11.6 Implementation Mappings Do Not Redefine Concepts
 - 11.7 Deployment Artefacts Do Not Redefine Implementation Artefacts
 - 11.8 Evidence Supports Claims
 - 11.9 Traceability Preserves Layering

- 11.10 Local Profiles Must Preserve Parent Meaning
- 11.11 Conceptual Constraints Summary
- 13. 12. Relationship to the Logical Architecture / PIM
 - 12.1 Overview
 - 12.2 Conceptual Elements Become Logical Elements
 - 12.3 Logical Architecture Preserves Conceptual Meaning
 - 12.4 Logical Architecture Remains Platform Independent
 - 12.5 Logical Architecture Supports Domain Profiles
 - 12.6 Logical Architecture Provides the Basis for Implementation Profiles
 - 12.7 Traceability from Part 1 to Part 2
 - 12.8 Boundary Between Part 1 and Part 2
- 14. 13. Conceptual Requirements
 - 13.1 Overview
 - 13.2 Conceptual Separation Requirements
 - 13.3 Classification Requirements
 - 13.4 Core Conceptual Element Requirements
 - 13.5 Runtime Plane Requirements
 - 13.6 Separation of Concerns Requirements
 - 13.7 Traceability Requirements
 - 13.8 Evidence Requirements
 - 13.9 Relationship to Logical Architecture / PIM Requirements
 - 13.10 Profile and Layering Requirements

From:

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

Permanent link:

<https://wiki.didosolutions.com/fxdemo/01-part/start?rev=1782871556>

Last update: **2026/06/30 19:05**

