

Part 4: Phase 0 Implementation Profile / PSM

[Go to The FX Demo Root](#)



Financial Systems Archetype

Foreign Exchange (FX) Demo

Abstract

Part 4 defines the Phase 0 implementation profile for the FX Demo Reference Architecture. It maps the logical profile from Part 3 onto the selected Phase 0 technology stack, including DDS, IDL, Python, containerised nodes, local runtime configuration, and initial implementation artifacts. Part 4 identifies how the Phase 0 implementation realizes the logical nodes, topics, messages, and runtime planes while preserving traceability back to the conceptual and logical architecture. It limits its scope to the Phase 0 platform-specific model and does not redefine the conceptual or logical architecture.

Contents

- [Foreword](#)
- [Introduction](#)
- [1. Scope](#)
- [2. Relationship to Parts 0, 1, 2, and 3](#)
- [3. Traceability to Parent and Source Architectures](#)
- [4. Phase 0 Implementation Profile Overview](#)
- [5. Phase 0 Implementation Principles](#)
- [6. Phase 0 Implementation Preconditions](#)
- [7. Phase 0 Technology Selection and Implementation Context](#)
- [8. Mapping from FX Logical Nodes to Implementation Artifacts](#)
- [9. Mapping from FX Logical Communication Endpoints to DDS Topics](#)
- [10. Mapping from FX Logical Information Structures to IDL and Generated Types](#)
- [11. Phase 0 Runtime Plane Realization](#)

- [12. Phase 0 Interaction Pattern Realization](#)
- [13. Phase 0 QoS Profile Model](#)
- [14. Phase 0 Repository and Build Artifact Model](#)
- [15. Phase 0 Developer Handbook Requirements](#)
- [16. Phase 0 Script and Execution Model](#)
- [17. Phase 0 Implementation Governance](#)
- [18. Phase 0 Implementation Traceability Model](#)
- [19. Relationship to Deployment, Testability, and Evidence Plan](#)
- [20. Phase 0 Implementation Rules and Constraints](#)
- [21. Phase 0 Implementation Requirements](#)

Contents

* [Part 4: Phase 0 Implementation Profile / PSM](#)

- [Foreword](#)
- [Introduction](#)
- [1. Scope](#)
- [2. Relationship to Parts 0, 1, 2, and 3](#)
- [3. Traceability to Parent and Source Architectures](#)
 - [3.1 Traceability to SIP-RA](#)
 - [3.2 Alignment with FDIS-RA](#)
 - [3.3 Traceability to Part 1 Conceptual Architecture](#)
 - [3.4 Traceability to Part 2 Distributed Node-Based Logical Architecture / PIM](#)
 - [3.5 Traceability to Part 3 FX Demo Logical Profile](#)
 - [3.6 Traceability to Phase 0 Implementation Source Material](#)
 - [3.7 Traceability to the Phase 0 Developer Handbook](#)
 - [3.8 Traceability Position](#)
- 6. [4. Phase 0 Implementation Profile Overview](#)
 - [4.1 Overview](#)
 - [4.2 Phase 0 Implementation Scope](#)
 - [4.3 Phase 0 Implementation Baseline](#)
 - [4.4 Phase-Specific Nature of the Profile](#)
 - [4.5 Relationship to the FX Demo Logical Profile](#)
 - [4.6 Relationship to Later Development Phases](#)
 - [4.7 Phase 0 Implementation Profile Summary](#)
- 7. [5. Phase 0 Implementation Principles](#)
 - [5.1 Overview](#)
 - [5.2 Implementation Artifacts Realize Logical Elements](#)
 - [5.3 Traceability before Code](#)
 - [5.4 Developer Handbook before Coding](#)
 - [5.5 Generated Artifacts before Manual Adaptation](#)
 - [5.6 Explicit Interfaces before Hidden Coupling](#)
 - [5.7 Runtime Plane Discipline in Implementation](#)
 - [5.8 Observable Behavior before Operational Evidence](#)
 - [5.9 Minimal Phase 0 Scope before Expansion](#)
 - [5.10 Phase 0 Implementation Principles Summary](#)
- 8. [6. Phase 0 Implementation Preconditions](#)
 - [6.1 Overview](#)

- 6.2 Part 3 Working Baseline
- 6.3 Phase 0 Developer Handbook
- 6.4 Repository Baseline
- 6.5 IDL and Generation Baseline
- 6.6 Logging and Exception-Handling Baseline
- 6.7 Traceability Baseline
- 6.8 No-Code-Before-Handbook Constraint
- 6.9 Phase 0 Implementation Preconditions Summary
- 9. 7. Phase 0 Technology Selection and Implementation Context
 - 7.1 Overview
 - 7.2 DDS as Phase 0 Communication Technology
 - 7.3 IDL as Phase 0 Data Definition Mechanism
 - 7.4 QoS Profiles as Phase 0 Communication Behavior Controls
 - 7.5 Generated Types as Phase 0 Implementation Data Artifacts
 - 7.6 Python as Phase 0 Implementation Language
 - 7.7 Scripts as Phase 0 Build and Execution Helpers
 - 7.8 Repository Structure as Phase 0 Implementation Organization
 - 7.9 Technology Selection Boundaries
 - 7.10 Phase 0 Technology Selection Summary
- 10. 8. Mapping from FX Logical Nodes to Implementation Artifacts
 - 8.1 Overview
 - 8.2 Node Mapping Principles
 - 8.3 Implementation Artifact Categories for Nodes
 - 8.4 FX Transaction Intake Node Mapping
 - 8.5 FX Validation Node Mapping
 - 8.6 FX Semantic Interpretation Node Mapping
 - 8.7 FX Contract State Node Mapping
 - 8.8 FX Cash-Flow Computation Node Mapping
 - 8.9 FX Policy and Release Node Mapping
 - 8.10 FX Audit and Provenance Node Mapping
 - 8.11 FX Health and Observability Node Mapping
 - 8.12 FX Registry and Coordination Node Mapping
 - 8.13 FX Replay and Reconstruction Node Mapping
 - 8.14 External Oversight Participant Mapping
 - 8.15 FX Logical Node Mapping Summary
- 11. 9. Mapping from FX Logical Communication Endpoints to DDS Topics
 - 9.1 Overview
 - 9.2 Topic Mapping Principles
 - 9.3 DDS Topic Naming Principles
 - 9.4 FX Transaction Intake Topic Mapping
 - 9.5 FX Validation Result Topic Mapping
 - 9.6 FX Semantic Interpretation Topic Mapping
 - 9.7 FX Contract State Topic Mapping
 - 9.8 FX Cash-Flow Obligation Topic Mapping
 - 9.9 FX Policy Decision Topic Mapping
 - 9.10 FX Release Package Topic Mapping
 - 9.11 FX Audit Record Topic Mapping
 - 9.12 FX Provenance Record Topic Mapping
 - 9.13 FX Node Status Topic Mapping
 - 9.14 FX Control Command Topic Mapping
 - 9.15 FX Replay Request Topic Mapping

- 9.16 FX Replay Result Topic Mapping
- 9.17 Endpoint-to-Topic Mapping Summary
- 12. 10. Mapping from FX Logical Information Structures to IDL and Generated Types
 - 10.1 Overview
 - 10.2 Information Structure Mapping Principles
 - 10.3 IDL Structure Naming Principles
 - 10.4 Generated Type Naming Principles
 - 10.5 FX Transaction Candidate Mapping
 - 10.6 FX Transaction Record Mapping
 - 10.7 FX Validation Result Mapping
 - 10.8 FX Semantic Assertion Mapping
 - 10.9 FX Contract State Mapping
 - 10.10 FX Cash-Flow Obligation Mapping
 - 10.11 FX Policy Decision Mapping
 - 10.12 FX Release Package Mapping
 - 10.13 FX Audit Record Mapping
 - 10.14 FX Provenance Record Mapping
 - 10.15 FX Node Status Mapping
 - 10.16 FX Control Command Mapping
 - 10.17 FX Command Acknowledgement Mapping
 - 10.18 FX Replay Request Mapping
 - 10.19 FX Replay Result Mapping
 - 10.20 Information-Structure-to-IDL Mapping Summary
- 13. 11. Phase 0 Runtime Plane Realization
 - 11.1 Overview
 - 11.2 Control Plane Realization
 - 11.3 Data Plane Realization
 - 11.4 Health and Observability Plane Realization
 - 11.5 Policy and Release Plane Realization
 - 11.6 Audit and Provenance Plane Realization
 - 11.7 Cross-Plane Implementation Relationships
 - 11.8 Runtime Plane Realization Summary
- 14. 12. Phase 0 Interaction Pattern Realization
 - 12.1 Overview
 - 12.2 Interaction Realization Principles
 - 12.3 Register FX Logical Node
 - 12.4 Report FX Node Status
 - 12.5 Intake FX Transaction Candidate
 - 12.6 Validate FX Transaction Candidate
 - 12.7 Interpret FX Transaction Semantics
 - 12.8 Establish or Update FX Contract State
 - 12.9 Compute FX Cash-Flow Obligations
 - 12.10 Request FX Policy Decision
 - 12.11 Release FX Information to Oversight Participant
 - 12.12 Record FX Audit and Provenance
 - 12.13 Replay or Reconstruct FX Information
 - 12.14 Interaction Pattern Realization Summary
- 15. 13. Phase 0 QoS Profile Model
 - 13.1 Overview
 - 13.2 QoS Profile Purpose
 - 13.3 QoS Profile Mapping Principles

- 13.4 Control Plane QoS Profile
- 13.5 Data Plane QoS Profile
- 13.6 Health and Observability Plane QoS Profile
- 13.7 Policy and Release Plane QoS Profile
- 13.8 Audit and Provenance Plane QoS Profile
- 13.9 QoS Profile Mapping Summary
- 13.10 QoS Boundaries and Non-Goals
- 16. 14. Phase 0 Repository and Build Artifact Model
 - 14.1 Overview
 - 14.2 Repository Organization Principles
 - 14.3 Source Directory Model
 - 14.4 IDL Directory Model
 - 14.5 Generated Code Directory Model
 - 14.6 Configuration Directory Model
 - 14.7 Script Directory Model
 - 14.8 Documentation Directory Model
 - 14.9 Traceability Artifact Model
 - 14.10 Build Artifacts
 - 14.11 Generated Artifacts
 - 14.12 Repository Baseline and Non-Baseline Material
 - 14.13 Repository and Build Artifact Summary
- 17. 15. Phase 0 Developer Handbook Requirements
 - 15.1 Overview
 - 15.2 Purpose of the Developer Handbook
 - 15.3 Handbook Approval before Coding
 - 15.4 File-Header Requirements
 - 15.5 Generated-File Header Requirements
 - 15.6 Naming Convention Requirements
 - 15.7 Coding Style Requirements
 - 15.8 Logging Convention Requirements
 - 15.9 Exception-Handling Requirements
 - 15.10 Traceability Convention Requirements
 - 15.11 Repository Workflow Requirements
 - 15.12 Script Usage Requirements
 - 15.13 Review Checklist Requirements
 - 15.14 Developer Handbook Summary
- 18. 16. Phase 0 Script and Execution Model
 - 16.1 Overview
 - 16.2 Script Role in Phase 0
 - 16.3 Prerequisite Checking
 - 16.4 Type Generation
 - 16.5 Build Preparation
 - 16.6 Node Execution
 - 16.7 Multi-Node Startup
 - 16.8 Multi-Node Shutdown
 - 16.9 Script Logging and Error Handling
 - 16.10 Script Approval and Baseline Status
 - 16.11 Script Model Summary
- 19. 17. Phase 0 Implementation Governance
 - 17.1 Overview
 - 17.2 Implementation Ownership

- 17.3 Implementation Versioning
- 17.4 Implementation Compatibility
- 17.5 Implementation Change Control
- 17.6 Generated Artifact Governance
- 17.7 Repository Governance
- 17.8 Developer Handbook Governance
- 17.9 Implementation Governance Summary
- 20. 18. Phase 0 Implementation Traceability Model
 - 18.1 Overview
 - 18.2 Part 3 FX Logical Element to Part 4 Implementation Artifact
 - 18.3 Implementation Artifact to Part 3 FX Logical Element
 - 18.4 Node Implementation Traceability
 - 18.5 Topic Implementation Traceability
 - 18.6 IDL and Generated Type Traceability
 - 18.7 Runtime Plane Implementation Traceability
 - 18.8 Script and Repository Traceability
 - 18.9 Developer Handbook Traceability
 - 18.10 Implementation Traceability Summary
- 21. 19. Relationship to Deployment, Testability, and Evidence Plan
 - 19.1 Overview
 - 19.2 Boundary Between Part 4 and Part 5
 - 19.3 Implementation Artifacts Prepared for Deployment
 - 19.4 Implementation Artifacts Prepared for Testability
 - 19.5 Implementation Artifacts Prepared for Observation
 - 19.6 Implementation Artifacts Prepared for Evidence
 - 19.7 Part 5 Handoff
 - 19.8 Relationship to Deployment, Testability, and Evidence Plan Summary
- 22. 20. Phase 0 Implementation Rules and Constraints
 - 20.1 Overview
 - 20.2 Implementation Artifacts Do Not Redefine Logical Elements
 - 20.3 DDS Topics Do Not Redefine Logical Communication Endpoints
 - 20.4 IDL Structures Do Not Redefine Logical Information Structures
 - 20.5 Generated Types Do Not Redefine Logical Information Structures
 - 20.6 Runtime Processes Do Not Redefine FX Logical Nodes
 - 20.7 Repository Layout Does Not Redefine Architecture
 - 20.8 Scripts Do Not Redefine Interaction Patterns
 - 20.9 Developer Handbook Does Not Replace the Implementation Profile
 - 20.10 Phase 0 Does Not Define Later Phase Implementation Profiles
 - 20.11 Deployment and Evidence Concerns Remain Outside Part 4
 - 20.12 Implementation Constraints Summary
- 23. 21. Phase 0 Implementation Requirements
 - 21.1 Overview
 - 21.2 Phase 0 Profile Requirements
 - 21.3 Technology Selection Requirements
 - 21.4 Logical Node Mapping Requirements
 - 21.5 Communication Endpoint Mapping Requirements
 - 21.6 Information Structure Mapping Requirements
 - 21.7 Runtime Plane Realization Requirements
 - 21.8 Interaction Pattern Realization Requirements
 - 21.9 QoS Profile Requirements
 - 21.10 Repository and Build Artifact Requirements

- [21.11 Developer Handbook Requirements](#)
- [21.12 Script and Execution Requirements](#)
- [21.13 Implementation Governance Requirements](#)
- [21.14 Implementation Traceability Requirements](#)
- [21.15 Part 5 Handoff Requirements](#)

Notes for Editors

This document should be maintained as a set of linked wiki pages rather than a single monolithic page. Each major section should have its own stable URI so it can be referenced from other wiki pages, exported documents, presentations, AI-assisted tools, and semantic models.

Do not rename pages once they have been cited externally unless a redirect or move plan is in place.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:

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

Permanent link:

<https://wiki.didosolutions.com/fxdemo/04-part/start>

Last update: **2026/07/18 12:33**

